User-Interface Guidelines

January 30, 2005

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

User-Interface Guidelines

January 30, 1996

Prepared for the
Data Systems Technology Division/Code 520
GODDARD SPACE FLIGHT CENTER

By
CTA INCORPORATED
6116 Executive Boulevard, Suite 800
Rockville, Maryland 20852

Under
Task Assignment 25-139
of
SEAS Subcontract 033336

Copies of this document may be obtained by writing to:

Sylvia Sheppard
Code 520
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

User-Interface Guidelines

Approval

Sylvia B. Sheppard Date
Head, Software and Automations Systems Branch
Code 522

Preface

The Data Systems Technology Division (Code 520) at the Goddard Space Flight Center (GSFC) continues to develop new systems employing both client-server and stand-alone workstation architectures. These are highly interactive, object-oriented groupware systems for which the user interface is very important.

Background and Purpose

In 1992, Code 520 published *Human-Computer Interface Guidelines* (DSTL-92-007) to provide human-factors guidance to designers and developers of user interfaces for both client-server and stand-alone systems. The 1992 guidelines were developed by Carlow International Incorporated and were subsequently reviewed by Teresa Bleser of Century Computing. She developed a set of recommendations for revision of the guidelines (Bleser, 1994).

The purpose of the present document is to provide an updated and expanded set of user-interface guidelines, building upon the core set of guidelines contained in the 1992 document and applying Dr. Bleser’s recommendations.

Scope

As before, the guidelines address the issues of user-interface (UI) design that are under the software developer's control. This document does not address operability and interface issues related to hardware characteristics (e.g., screen resolution, phosphor characteristics, workspace), which are not controlled by the developer. Depending on the development environment, the resolution of some issues will be defined by the user-interface development package or toolkit. The guidelines provided here are phrased independently of any of the user-interface toolkit style guides, unless so noted in the detailed references. In all cases, the style guide of a particular development system takes precedence over these guidelines.

Key Definitions and Examples

This document presents user-interface design *principles* and *guidelines*, not design rules, standards, requirements, or specifications. A design principle is a broad statement of a general characteristic that user interfaces should have. For example, the principle of consistency holds that a user interface should have a consistent “look and feel,” but it does not indicate how to achieve consistency.

Guidelines take principles to the next lower level, but are still quite generic. For example, a guideline under the principle of consistency is to allow users to predict system actions based on previous experience with other system actions. This focuses on achieving consistency through consistent system behavior, but it could be implemented in many different ways.

At the next lower level, to ensure consistency across developers, design rules must be developed by each project, as explained in the Introduction to this document (Section 2.0). A *design rule* specifies how to implement a particular feature of the user interface. For example, to achieve a user interface that appears consistent from screen to screen, a particular project might decide that "iconic buttons should be 10 pixels from the fields above and 10 pixels from the bottom of the window" (Campbell, 1995). Another project might decide on a different amount of spacing between buttons and fields. Design rules, then, are standards which cannot be stated generically because they are likely to vary from system to system.

Benefits and Limitations

The guidelines are intended to help software developers generate user interfaces that support the end user's tasks and functional requirements. Such tasks and functions include database search, archival data entry, project scheduling and planning, project resource management, and information distribution and control. To use the guidelines effectively as a basis for a project style guide, the development team must be thoroughly familiar with the end users' tasks. The guidelines cannot substitute for knowledge of the information requirements associated with decision points embedded in the users' tasks. This document assumes that developers have detailed user requirements available to them through other sources, such as cognitive task analyses and frequent, direct interaction with end users.

Document Organization:

This document is organized into eight major sections, followed by a glossary, a list of references, an appendix on user-interface design methods, and a detailed index.

Section 1.0 -- Introduction: How to Use the Guidelines recommends an approach to using the contents of this document.

Section 2.0 -- User-Centered Design Principles and Guidelines offers high-level principles as general guidance in designing a user interface and includes guidelines that take the principles to lower levels.

Section 3.0 -- Guidelines for Basic Interface Components presents design guidance on such UI elements as cursors, fonts, text fields, pushbuttons, and dialog boxes. This section includes guidelines on display control and sequence control.

Section 4.0 -- Guidelines for Screen Layout and Design focuses on orderly design of visual displays to support the exchange of information between the user and the computer. Key topics include general layout, alphanumerics, and emphasis techniques.

Section 5.0 -- Guidelines for Interaction Styles provides guidance on the design of alphanumeric and graphical dialogs. The following styles of interaction are addressed: fill-in forms, question-and-answer dialogs, menus, direct manipulation. This section also presents alternative techniques to support browsing.

Section 6.0 -- Guidelines for Window Management emphasizes the need for design to minimize the time users must spend in window configuration and management. Major topics include general window appearance and behavior ("look and feel").

Section 7.0 -- Guidelines for Visual Coding Techniques addresses issues in the use of color, brightness, flash coding, line coding, and other methods for supporting the user's rapid detection and discrimination of displayed data.

Section 8.0 -- Guidelines for Feedback to the User suggests ways to improve communication with the user through prompts, general guidance messages, cautions, and warnings. Guidelines on the design of error messages and on-line help are included in this section.

The *Glossary* defines key terms used in the body of the document. Terms italicized in the text are defined in the Glossary.

The *References* provide detailed information on the authorship and publication of sources cited for the guidelines and for terms in the glossary.

Appendix A: User-Interface Design and Evaluation Methods presents a brief overview of the user-interface-design process as it is practiced in Code 520.

The hierarchical *Index* provides a means of locating topics in the body of the document.

Numbering Scheme:

Each section is given a high-level number ending in a zero (e.g., 1.0, 2.0...8.0). Key topics in each section carry the zero down to one digit (e.g., 1.1, 2.3, 3.5). At the topic level, a brief textual introduction is typically provided.

Guidelines are presented at the next level and all lower levels (e.g., 5.1.2, 5.1.2.1, 5.1.2.2, 5.1.2.2.1). The longer the number associated with a guideline, the more detailed is the guideline. Guidelines are not listed in order of importance, but they are listed according to level of detail.

References for Guidelines:

Supporting references for the introductory material and the guidelines are provided at the end of each section, using the numbering scheme for that section. Full publication information is provided in the References section, which appears after the Glossary.

Abstract

This document updates and expands upon the earlier NASA/Goddard Code 520 *Human-Computer Interface Guidelines* (DSTL-92-007). The guidelines are intended for use by development teams in generating project-specific style guides to ensure consistency across developers of particular user interfaces. As such, the guidelines represent an intermediate stage of design guidance, between high-level user-interface design principles and specific design rules. The guidelines have been selected for their applicability to both client-server and stand-alone workstation environments.

Keywords: User-interface design
Guidelines on “look and feel”
Interaction styles
Window management
Visual coding techniques
Feedback to the user

Acknowledgements

The developers of the guidelines wish to acknowledge the prior work by Carlow International Incorporated, upon which the current document is based. We also want to express our appreciation to Dr. Teresa Bleser for her expert evaluation of the original guidelines and to the following members of the Guidelines Working Group, who gave generously of their time and expertise during the process of revision:

Chris Burkhard
Rachel Campbell
Melana Clark
Roger Clason
Johnny Medina
Sylvia Sheppard
John Stachniewicz
Mark Stephens
Dana Uehling

Special thanks to both Dana Uehling and Rachel Campbell for providing sample illustrations and to Dana for her detailed review of earlier drafts.

DCN Control Page

Use this page to record Document Change Notice (DCN) changes to this document. Since one DCN may define more than one change, use as many lines as necessary to record the section(s) changed by each DCN.

DCN No.	DCN date	Section(s)	Date added	Initials

Table of Contents

Table of Contents (Cont’d)

Table of Contents (Cont'd)

Table of Contents (Cont'd)

Table of Contents (Cont'd)

List of Figures

List of Tables

Acronyms and Abbreviations

ANSI	American National Standards Institute
CHI	computer-human interface
dd	placeholder for 2-digit number indicating the day of the month (e.g., 01 for the first day, 30 for the thirtieth day)
DSTL	Data Systems Technology Laboratory
ESC	the escape key on a computer keyboard
GL	guideline
GUI	graphical user interface
HCI	human-computer interface
HFE	human factors engineering
Hz	Hertz (cycles per second)
ID	identification
ISO	International Organization for Standardization
LTM	long-term memory
mm	placeholder for 2-digit number indicating the month of the year (e.g., 01 for January; 12 for December)
MMI	man-machine interface (obsolete term)
NASA	National Aeronautics and Space Administration
OSF	Open Software Foundation
STM	short-term memory
UI	user interface
USI	user-system interface

yy	placeholder for a 2-digit number indicating the year (e.g., 95 for 1995)

1.0 Introduction: How to Use the Guidelines

The guidelines on user-interface design found in this document are necessarily generic. They are meant to apply across a wide range of applications. In order to do so, they must be worded at a relatively high level. The guidelines are a starting point for the development of an application-specific style guide. They do not, in themselves, represent a style guide that can be handed to developers with any expectation that a consistent, usable interface will emerge.

What each project needs to do is to select the guidelines that are meaningful in the context of the user interface to be developed. Next, each guideline must be developed further, to the point that a clear, specific design rule is defined. The full set of design rules then makes up the project's style guide.

1.1 Developing a Style Guide

The process involved in developing a style guide includes the following steps, as recommended by Bleser (1994):

- Identify relevant guidelines. From the overall set of guidelines, select those that pertain to the application under development.

- Narrow down the subset of pertinent guidelines. The subset of guidelines selected in the first step may include some that conflict. The choice of which guidelines to retain may be based on relative importance or impact, given constraints of time and budget.

- Develop design rules from the guidelines. A process of translation is required to move from high-level guidelines to specific design rules. One guideline may require a whole set of design rules. If a guideline states that displays should be formatted consistently, for example, a set of design rules would be needed to specify the location of such display features as menu titles, icon labels, dialog boxes, and error messages. Design rules take the guidelines down to a concrete, highly specific level.

Because a particular guideline can be translated in numerous ways, translation requires designers to define interface components, application components, and constraints that must be met.

- Document and distribute the design rules. Each member of the design and development team needs unambiguous guidance on the rules that are to be followed. The goal of collecting the design rules in a style guide is to encourage consistency in the "look and feel" of the application. For an example of a style guide see *Client-Server Systems User Interface Style Guide* (Campbell, 1994).

- Allow for reasonable exceptions. As the design rules are applied during design, some rules may turn out to be in conflict with others or may simply be inapplicable due to design constraints. In such cases, the group can agree to make exceptions, record and distribute the rationale for any exceptions, and perhaps revise the rules in question.

Occasionally, a conflict may arise between guidelines. Sometimes the development team can decide which guideline to accept simply by considering whether one or the other is more appropriate for their application. When the answer is not clear, however, the team can use a more formal decision-making process, according to the following steps:

a. Identify the attributes of user performance that may be affected by the conflicting guidelines (e.g., color discrimination, target detection, speed of response).

b. Weight the importance of those attributes for overall system performance. These weightings are likely to vary from project to project.

c. Rate the conflicting guidelines for their expected effect on each performance outcome. The rating scale should have at least three alternatives (e.g., high (1), moderate (2), low (3)) but should always have an odd number of alternatives so that a mid-point is defined.

d. Multiply ratings by weights and sum the products. Select the guideline with the higher total.

1.2 From Guidelines to Design Rules

Several examples follow to illustrate the transition from guideline to design rule:

- Guideline Example For Buttons: When the same buttons are used for different windows, consistently place them in the same location and keep related buttons together.

- Transition Questions: Which buttons are involved? Are there any related buttons? Where should these buttons be placed in this application?

- Sample Design Rule: Place "window-level" buttons at the bottom of the window.

- Guideline Example For Labels: Label each data field to inform users of entries to be made. Keep labels close to associated data fields; separate them by at least one space. For more clarity, employ additional cues in a field label or in the field itself.

- Transition Questions: How can we aid users in knowing where the label ends and the entry field begins? How many spaces should there be between the label and the data?

- Sample Design Rule: Use three-dimensional (3D) shading to delineate a data field.

- Guideline Example For Graphical Aids: Provide graphical or textual aids to assist users in maintaining their orientation within the underlying menu structure.

- Transition Questions: Will a graphical or textual aid benefit our users? If the aid is to be graphical, what should it include? Should the aid be displayed continuously or not?

- Sample Design Rule: At user request, display a small schematic of the entire menu structure. Use the schematic provided by the menu project manager. As the user proceeds through the menu structure, highlight the path taken in yellow.

Design rules should be specific enough that different developers will produce exactly the same features when applying them. For this reason, they should be pre-tested to ensure that developers will agree in their interpretation. There should be little room for a variety of interpretations.

References

1.0	Bleser (1994); Campbell (1994)
1.1	Bleser (1994); Campbell (1994)

2.0 User-Centered Design Principles and Guidelines

User productivity can be enhanced by providing consistent and comprehensible displays, flexibility to change or structure a system, informative feedback, error tolerance, and reduced demands on short-term memory–all of which give the user a sense of competence, mastery, and control over the system.

The following principles are offered as general guidance in designing a user interface. These principles represent a condensation of the many general principles articulated in the literature on user-interface design. (Principles are given a second-level identifier, such as 2.1; guidelines are given third and fourth-level identifiers.)

2.1 Maintain Consistency in Look and Feel

Consistent visual appearance and consistent response to user input are required throughout the user interface. As illustrated in Figure 2-1, interface characteristics should be uniform and familiar, with consistent sequences of actions in similar situations. Terminology must be used consistently to avoid confusing the user. A user interface becomes intuitive by consistently meeting users' expectations.

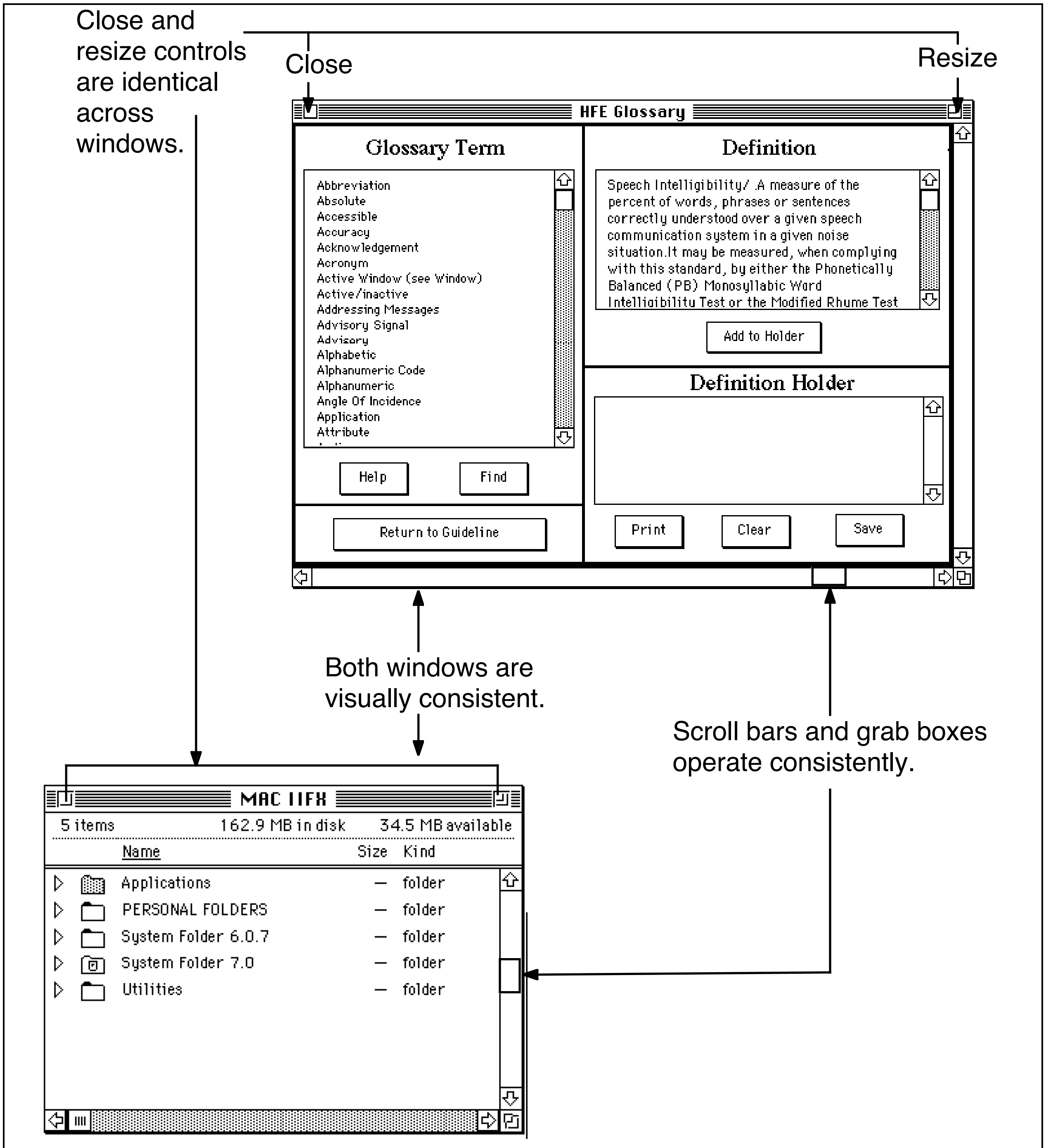
2.1.1 Consistent Interface Characteristics

Allow the user to build-up expectations and predict system actions based on the system's performance of other actions.

2.1.1.1 Permit the user to take the general knowledge and skills learned in one system and transfer them to another like it, without requiring extensive learning and training exercises.

Positive transfer may be based on any or all of the following:

- Analogy with manual methods (e.g., a file cabinet for a collection of files).
- Experience with similar systems.
- Previous experience in life or culture (e.g., red = danger).
- Experience with this system's consistent "look and feel".



Close and resize controls are identical across windows.

Both windows are visually consistent.

Scroll bars and grab boxes operate consistently.

Abbreviation
Absolute
Accessible
Accuracy
Acknowledgement
Acronym
Active Window (see Window)
Active/inactive
Addressing Messages
Advisory Signal
Advisory
Alphabetic
Alphanumeric Code
Alphanumeric
Angle Of Incidence
Application
Attribute

Speech Intelligibility/ A measure of the percent of words, phrases or sentences correctly understood over a given speech communication system in a given noise situation.It may be measured, when complying with this standard, by either the Phonetically Balanced (PB) Monosyllabic Word Intelligibility Test or the Modified Rhume Test

5 items 162.9 MB in disk 34.5 MB available

Name	Size	Kind
Applications	—	folder
PERSONAL FOLDERS	—	folder
System Folder 6.0.7	—	folder
System Folder 7.0	—	folder
Utilities	—	folder

Figure 2-1. Consistent Interface Characteristics

2.1.1.2 Maintain consistency in the following design areas:

- Display:
  - Icon design and meaning
  - Title field location
  - Menu bar location
  - Message location
  - Cursor shape and function
  - Cursor home position
  - Field delimiters
  - Color meanings
  - Data entry prompt
- Labeling terminology
- System Control:
  - Command Terminology
  - Command meanings
  - Editing procedures
  - Function keys
  - Command keys
- Abbreviations
- Mnemonics
- Acronyms
- Alarms and Warnings
- Visual Coding

2.1.1.3 Occasional departures from consistency may be necessary to support user task performance or convenience. If such departures are necessary, try to minimize the extent of inconsistency with the rest of the user interface.

2.2 Provide Shortcuts and Flexibility

The experienced user needs the means to go directly to specific locations in the UI structure. The UI should be flexible enough to accommodate different user styles of performing tasks.

2.2.1 Number of Interactions

Limit the number of interactions a frequent user must perform. Based on their knowledge, skill, and experience, users should have the flexibility to change or structure a system to suit their particular requirements.

2.2.2 Self-Pacing

Ensure that the pacing of inputs is controlled by the user.

2.2.3 Keyboard Commands

Provide keyboard commands for use by more experienced users, as alternatives to cursor pointing and selection.

2.2.4 Loading on User Memory

To support job performance, mnemonics, codes, special or long sequences, and detailed instructions should be kept to a minimum. For example, use a one-letter mnemonic such as F for File and O for Format.

2.2.5 Log-On/Off

Support ease of logging-on and logging-off by:

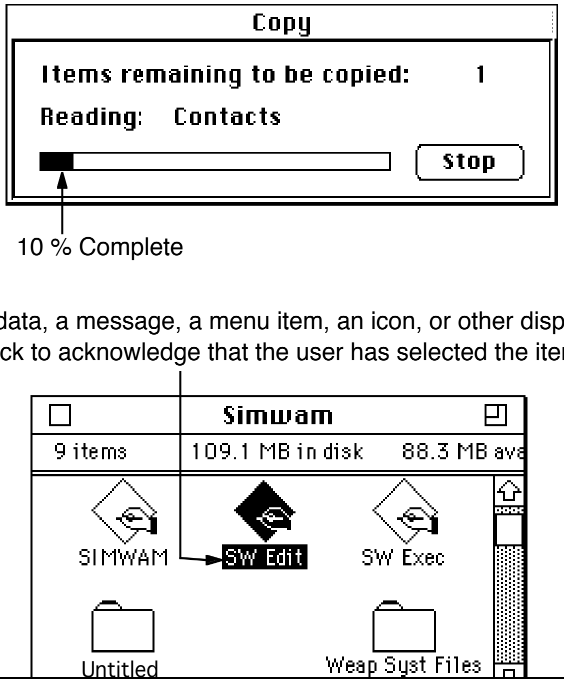
- Providing the user the means to log-on and log-off by a single action.
- Prior to accepting log-off, informing the user if there are pending actions.
- For automatic log-off, providing an audible signal prior to log-off.
- Permitting the user to save the contents of the task document before log-off.

2.3 Present Informative Feedback

Learning and user confidence result from feedback. Feedback informs the user when processing is in progress (Figure 2-2) or when the system has completed a request. Feedback also indicates user selection of a displayed element, such as a menu option.

If a process is time-consuming and causes the screen and input devices to be locked, display and update a progress message.

Highlight data, a message, a menu item, an icon, or other display struct as feedback to acknowledge that the user has selected the item.



Copy

Items remaining to be copied: 1

Reading: Contacts

Figure 2-2. Feedback Messages

2.3.1 Automatic Validation

For data processing, provide automatic validation to check for entries of correct content and/or format. If incorrect data are entered, a message should be generated requesting a revised entry.

2.3.2 Temporary Deferral

For the user who wants to defer a required data item, provide a special symbol to be entered by the user, indicating that the item has been temporarily omitted and not ignored. Upon a request to process entries that include deferred data items, inform the user of the omission(s) and permit immediate entry of missing items, or allow for further deferral.

2.3.3 Clarity and Brevity

Present information in a manner that is understandable and concise.

2.3.3.1 When displaying text for user guidance, use simple and clear wording.

2.3.3.2 Begin sentences with the main topic. Keep sentences short and simple.

2.3.3.3 Use distinct words. Avoid contractions or combined forms.

2.3.4 Rules of Message Composition

2.3.4.1 Use affirmative rather than negative command statements.

2.3.4.2 Use active voice rather than passive voice.

2.3.4.3 Phrase a sequence of events in corresponding word order.

2.3.4.4 Display a series of related items in a list, not continuous text (Table 2-1).

Table 2-1 List of Related Items

Not Good	Better
Create a new logon by entering a user ID, entering your full name, entering the organization code, and pressing the Save button.	To Create A New Logon: • Enter a user ID • Enter your full name • Enter the organization code • Press the Save button

2.3.4.5 Base the level of detail on the user's knowledge and experience. Examples of messages for inexperienced users are illustrated in the Better column of Table□2-2.

Table 2-2 Examples of Messages for the Novice

Not Good	Better
Position the cursor on Save and click.	Click on the Save button.
Date:	Date mm/dd/yy: _ _/_ _/_ _
Press UP/DOWN arrows to move up or down.	To move up: Press UP arrow To move down: Press DOWN arrow
Pressing ESC will cause you to exit.	To exit, press the ESC button.

2.3.5 Message Location

Provide a consistent location for messages, such as a designated line at the bottom of the screen or a window.

2.3.5.1 Make the message distinct from other displayed information by using techniques such as highlighting, reverse video, or different fonts.

2.3.5.2 Display messages in mixed-case.

2.4 Design for Recovery from Error

When an error occurs, tell the user what the error is and how to correct it.

2.4.1 Undo Function

When possible, permit easy reversal of actions. It benefits the user to know that an error can be "undone."

2.4.1.1 Provide an undo command that immediately enables the user to reverse the previous control, entry, change, or delete action.

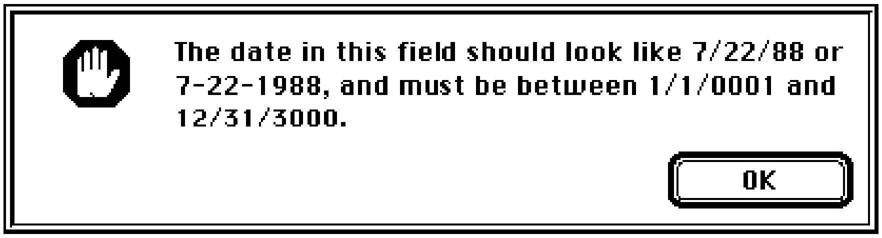
2.4.1.2 Make an undo action reversible. A second undo action should reinstate whatever was just undone.

2.4.2 Meaningful Error Messages

2.4.2.1 Present error messages that state the nature of the error and provide possible solutions. As illustrated in Figure 2-3, the messages should be brief and worded in terms of the task.

2.4.2.2 Enable the user to inquire about the error in more detail and/or request additional information on the operation in progress.

When an entry is in error, provide a feedback message.



The date in this field should look like 7/22/88 or 7-22-1988, and must be between 1/1/0001 and 12/31/3000.

Figure 2-3. Brief, Task-Related Error Message

2.4.3 Backup Function

Make it possible to backup in a transaction sequence to correct errors and/or make changes.

2.4.4 Minor Deviations

Accept minor deviations. Under some circumstances, there are acceptable deviations, such as equating "exit" with off, log-off, quit, or bye.

2.5 Reduce Memory Demands

Human capacity for information processing has its limitations. Short-term memory provides a limited mental "scratchpad" for information processing and problem solving. Too many facts and decisions may overload short-term memory.

2.5.1 Length and Complexity

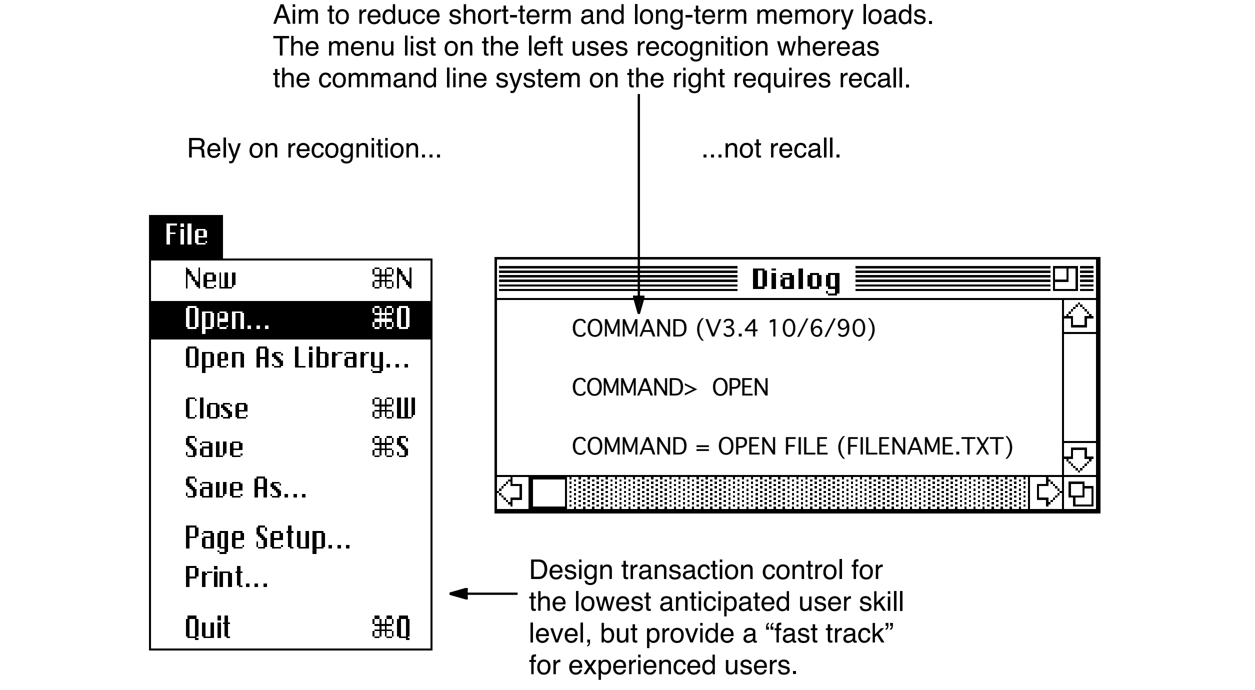
Do not require the user to remember lengthy lists of codes and complex command strings.

2.5.2 Amount of Input Activity

Require fewer input activities to increase user productivity.

2.5.3 Selection through Recognition

Provide selection(s) from a list of choices. This eliminates memorization, structures decision-making, and does away with typographical errors. In Figure 2-4, selecting from a pull-down menu is an example of recognition, which places little demand on short-term memory, as compared to unaided recall of command strings.



Aim to reduce short-term and long-term memory loads. The menu list on the left uses recognition whereas the command line system on the right requires recall.

Rely on recognition...

...not recall.

```
COMMAND (V3.4 10/6/90)

COMMAND>  OPEN

COMMAND = OPEN FILE (FILENAME.TXT)
```

Design transaction control for the lowest anticipated user skill level, but provide a "fast track" for experienced users.

Figure 2-4. Recognition as a Memory Aid

2.6 Design for Task Relevance

Present information that pertains to the user's task. Any arrangement of items on the screen (in menus, lists, tables, etc.) should reflect task requirements.

2.6.1 Task-Related Capabilities

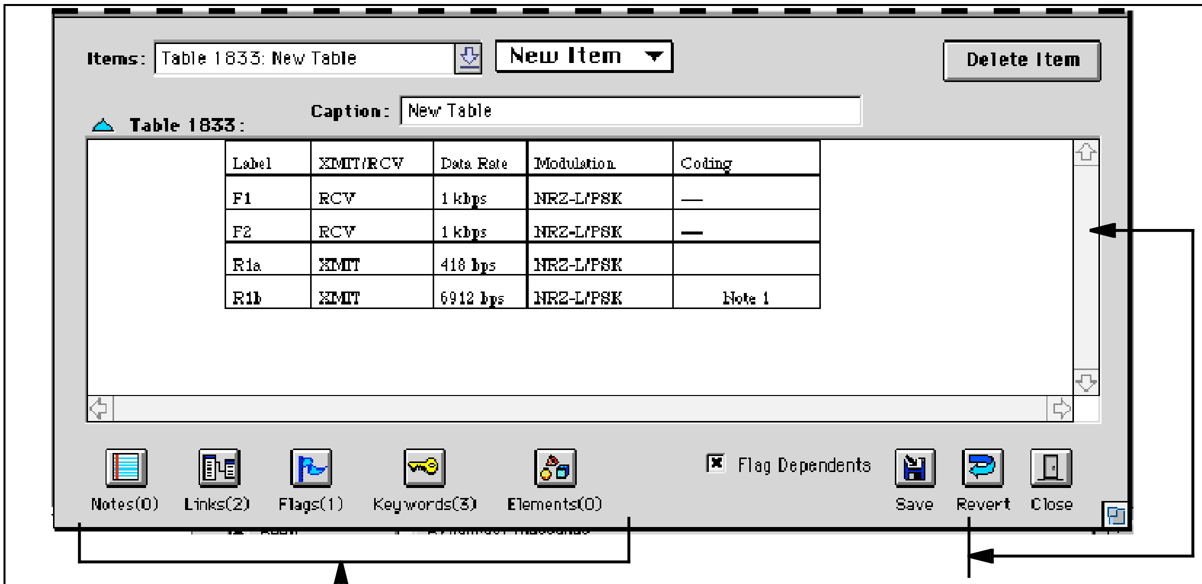
Use system capabilities to support the user's task performance. For example, use color, highlighting techniques, and graphics only to enhance user task accomplishment.

2.6.1.1 Make the design focus on the task, not on what the user must do with the hardware and software to accomplish the task.

2.6.1.2 Use dialog techniques that reflect the user's view and conception of what needs to be done.

2.6.2 Familiar Terms

Use terminology that is familiar to the user. Abbreviations, icons, mnemonics, codes, and acronyms should stem from specific job-related terminology or a known logic (Figure 2-5). Avoid the technical language of designers and programmers.



Items: Table 1833: New Table

Caption: New Table

Table 1833:

Label	XMIT/RCV	Data Rate	Modulation	Coding
F1	RCV	1 kbps	NRZ-L/PSK	—
F2	RCV	1 kbps	NRZ-L/PSK	—
R1a	XMIT	418 bps	NRZ-L/PSK	
R1b	XMIT	6912 bps	NRZ-L/PSK	Note 1

☒ Flag Dependents

Icons and images are based on operational or functional metaphors

Simple navigational controls and labels are apparent as to function and meaning

Figure 2-5. Job-Related Terminology

2.7 Aid Orientation and Navigation

Provide orientation aids and instructions to help users maintain a sense of where they are in the system, what they can do, and how they can get out (Figure 2-6).

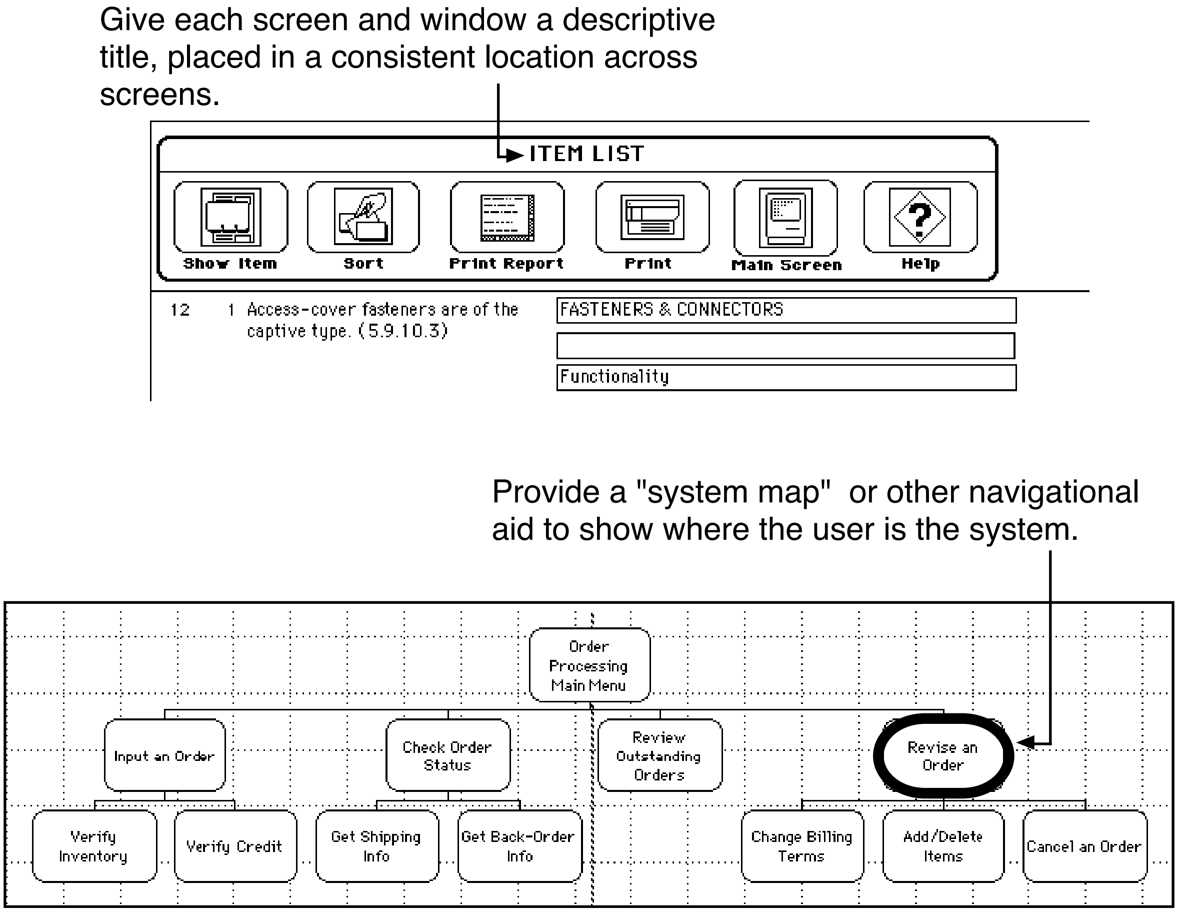
2.7.1 Descriptive Title

Include a descriptive title placed in a consistent location on each screen, window, and menu.

2.7.1.1 Provide a clear, distinctive, and short title that reflects the content and purpose of the screen. Avoid words such as FORM and SCREEN and connecting words like "of" in the title.

2.7.1.2 Center the title on the screen.

2.7.1.3 Display the screen title in capital letters or mixed case.



Give each screen and window a descriptive title, placed in a consistent location across screens.

ITEM LIST

12 1 Access-cover fasteners are of the captive type. (5.9.10.3)

FASTENERS & CONNECTORS

Functionality

Provide a "system map" or other navigational aid to show where the user is the system.

Figure 2-6. Orientation/Navigation Aids

2.7.2 Screen Identifiers

Use a numbering scheme to identify the currently displayed page and the total number of pages in a multipage display (e.g., 2 of 20).

2.7.2.1 Locate screen identifiers consistently, preferably to the right of the screen title.

2.7.3 System Map

When appropriate, provide a “system map” to show users where they are in the system (Figure 2-6).

2.7.4 Anticipation of User Actions

Anticipate possible user actions. In preparation for possible user actions provide the user with the following:

- A uniform starting point for control entries.
- Available transaction options.
- Task-relevant menus and consistently located control options.
- Editable fields that are distinct from non-editable fields.
- Default values, displayed automatically in the appropriate data fields.
- User-interrupt, continue and abort functions.
- A simple means of navigating between windows.

2.7.5 Exiting

Provide the user a means to log-off by a single action (e.g., menu option, command input).

2.7.5.1 Inform the user of any pending actions that will be lost upon log-off.

2.7.5.2 Permit the user to exit a file without saving changes, but require a confirmation to exit without saving changes.

2.7.5.3 Give the user a means to stop interacting with any type of file by a single exiting action (e.g., menu option, command input).

2.8 Provide On-Line Help

Provide users with on-line help that can be entered whenever needed (Figure□-7). Tailor help to the task context and requirements. Presenting the entire user's manual in response to a help query will only frustrate the user. Effective help can serve as a data-protection resource.

2.8.1 Multi-Level Help

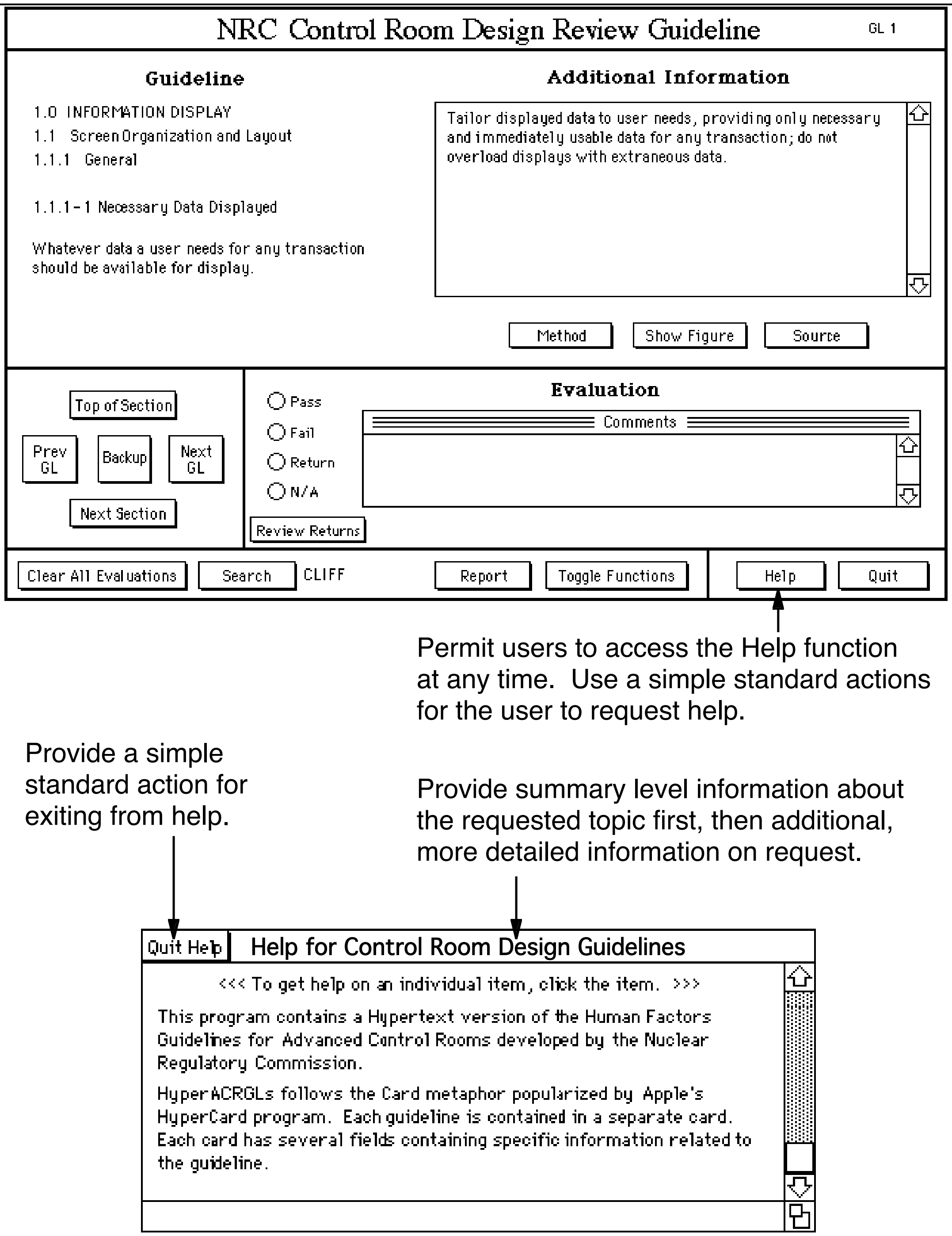
Provide multi-level help, beginning with summary information and providing more detailed explanations on request (Figure 2-7).

2.8.2 Access to Help

Permit the user to enter help at any point (Figure 2-7).

2.8.3 Request for Help

Use a simple, standard action for the user to request help (Figure 2-7).



NRC Control Room Design Review Guideline

GL 1

Guideline

1.0 INFORMATION DISPLAY
1.1 Screen Organization and Layout
1.1.1 General

1.1.1-1 Necessary Data Displayed

Whatever data a user needs for any transaction should be available for display.

Additional Information

Tailor displayed data to user needs, providing only necessary and immediately usable data for any transaction; do not overload displays with extraneous data.

Permit users to access the Help function at any time. Use a simple standard actions for the user to request help.

Provide a simple standard action for exiting from help.

Provide summary level information about the requested topic first, then additional, more detailed information on request.

Help for Control Room Design Guidelines

<<< To get help on an individual item, click the item. >>>

This program contains a Hypertext version of the Human Factors Guidelines for Advanced Control Rooms developed by the Nuclear Regulatory Commission.

HyperACRGLs follows the Card metaphor popularized by Apple's HyperCard program. Each guideline is contained in a separate card. Each card has several fields containing specific information related to the guideline.

Figure 2-7. Help Function

2.8.4 Help Browser

Permit the user to browse help topics.

2.8.5 Context-Sensitive Help

Tailor available help to task context and requirements. (See Section 8 for further detail).

2.8.6 Automatic Help

Activate the help function automatically (or offer help) when the user is making repeated errors.

2.8.7 Return to Task

Provide an easy means of returning to the task after accessing help (Figure 2-7).

2.9 Maintain a User-Centered Perspective

Focusing on user requirements is the key to maintaining a user-centered perspective. Every element of the design should be traceable to user requirements. From this perspective it is counterproductive to introduce "bells and whistles" just because it is technically possible to implement them. Capabilities not needed by the user will remain unused or, worse, may result in confusion.

2.9.1 User in Control

Keep the user in charge. A system that gives users a sense of control and responds to their actions builds trust and acceptance.

2.9.2 Decision Assistants

Design any required decision aids as assistants to (not replacements for) the user's flexible decision-making capabilities.

References

2.0	Norman and Draper (1986); Shneiderman (1992); Mayhew (1992); Hix and Hartson (1993); Carlow (1992); Smith and Mosier (1986); Avery and Bowser (1992); Apple Computer (1987); Open Software Foundation (1991); Mullet (1995)
2.1	Shneiderman (1992); Smith and Mosier (1986); Apple Computer (1987); Open Software Foundation (1991); Marcus (1992); Mullet (1995)

2.1.1	Carlow (1992)
2.1.1.1	Carlow (1992)
2.1.1.2	Carlow (1992)
2.1.1.3	Grudin (1989)
2.2	Smith and Mosier (1986); Shneiderman (1992)
2.2.1	Shneiderman (1992)
2.2.2	Smith and Mosier (1986)
2.2.3	Shneiderman (1992)
2.2.4	Smith and Mosier (1986); Shneiderman (1992)
2.2.5	Smith and Mosier (1986); Galitz (1993)
2.3	Shneiderman (1992)
2.3.1	Smith and Mosier (1986)
2.3.2	Smith and Mosier (1986)
2.3.3	Smith and Mosier (1986); Galitz (1993)
2.3.3.1	Smith and Mosier (1986); Galitz (1993)
2.3.3.2	Smith and Mosier (1986); Galitz (1993)
2.3.3.3	Smith and Mosier (1986); Galitz (1993)
2.3.4	Smith and Mosier (1986); Galitz (1993)
2.3.4.1	Smith and Mosier (1986); Galitz (1993)
2.3.4.2	Smith and Mosier (1986); Galitz (1993)
2.3.4.3	Smith and Mosier (1986); Galitz (1993)
2.3.4.4	Smith and Mosier (1986)
2.3.4.5	Mayhew (1992)
2.4	Shneiderman (1992); Galitz (1993)
2.4.1	Smith and Mosier (1986); Mayhew (1992)
2.4.1.1	Smith and Mosier (1986); Mayhew (1992); Shneiderman (1992)
2.4.1.2	Carlow (1992); Smith and Mosier (1986)
2.4.2	Shneiderman (1992); Mayhew (1992); Galitz (1993)
2.4.2.1	Smith and Mosier (1986); Mayhew (1992); Galitz (1993); Shneiderman (1992)
2.4.2.2	Smith and Mosier (1986)
2.4.3	Mayhew (1992); Apple Computer (1987); Open Software Foundation (1991); Smith and Mosier (1986); Avery and Bowser (1992)
2.4.4	Galitz (1993)
2.5	Shneiderman (1992); Mayhew (1992)
2.5.1	Mayhew (1992)
2.5.2	Smith and Mosier (1986)
2.5.3	Smith and Mosier (1986)
2.6	Smith and Mosier (1986); Carlow (1992); Galitz (1993)
2.6.1	Smith and Mosier (1986)
2.6.1.1	Carlow (1992)
2.6.1.2	Carlow (1992)
2.6.2	Smith and Mosier (1986)
2.7	Carlow (1992); Mayhew (1992); Galitz (1993)
2.7.1	Carlow (1992); Galitz (1993)
2.7.2	Carlow (1992)

2.7.2.1	Smith and Mosier (1986)
2.7.3	Carlow (1992)
2.7.4	Carlow (1992); Mayhew (1992); Galitz (1993)
2.7.5	Carlow (1992)
2.7.5.1	Carlow (1992)
2.7.5.2	Carlow (1992)
2.7.5.3	Carlow (1992)
2.8	Carlow (1992); Smith and Mosier (1986)
2.8.1	Carlow (1992); Smith and Mosier (1986)
2.8.2	Carlow (1992)
2.8.3	Carlow (1992); Smith and Mosier (1986)
2.8.4	Carlow (1992); Smith and Mosier (1986)
2.8.5	Carlow (1992); Smith and Mosier (1986)
2.8.6	Carlow (1992)
2.8.7	Carlow (1992)
2.9	Carlow (1992); Shneiderman (1992)
2.9.1	Shneiderman (1992)
2.9.2	Woods and Roth (1988)

3.0 Guidelines for Basic User-Interface Components

Basic components of alphanumeric user interfaces include such elements as text fields, tables, and lists. Graphical user interfaces add lines, shapes, pushbuttons, icons, and dialog boxes. Both types of user interfaces incorporate one or more cursors, and labeling is important in both. The key design challenge is to select and integrate all UI components into a seamless whole.

3.1 Cursors

Cursors enable the user to move the focus of input or attention within a display (Figure□3-1). Cursor control should provide fast movement and accurate placement.

A placeholding cursor should be easy to see and should not interfere with detection of any adjacent symbol or character. A second, pointing cursor should be visually distinct from the placeholding cursor.

3.1.1. Placeholding Cursor

On the screen, a cursor is typically positioned for quick and easy start of the keying process. The placeholding cursor is a mark on the display, indicating the current position for attention.

3.1.1.1 Place the cursor at the first data entry field to which the user must provide input, and advance the cursor to the next data field when the user has completed entry in the current field.

3.1.1.2 If the placeholding cursor blinks, the default rate should be 3 Hz. User selectable blink rate should be between 3 - 5 Hz.

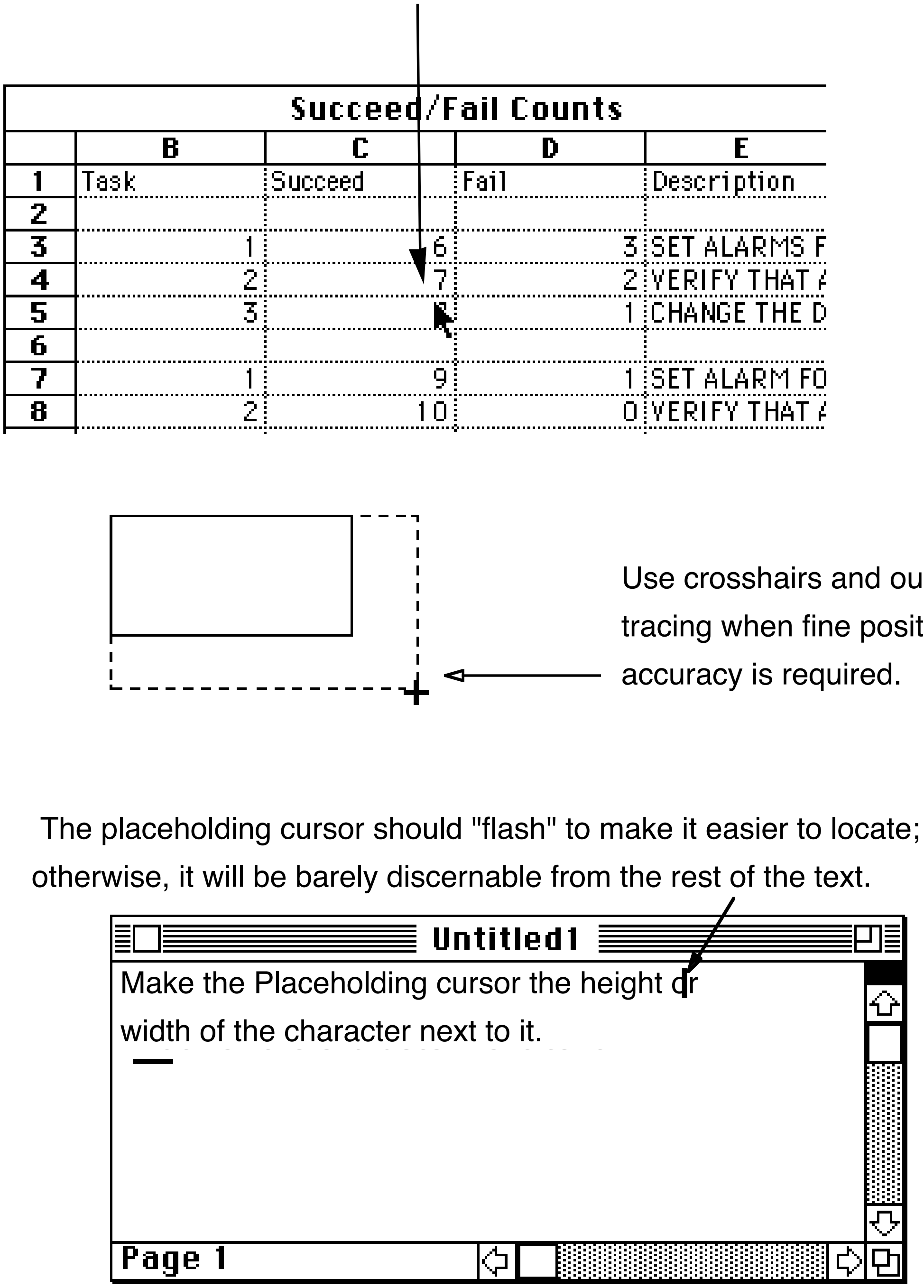
3.1.1.3 Use only one placeholding cursor in each window in which the user is entering alphanumeric characters.

3.1.1.4 Make a placeholding cursor the height or width of the alphanumeric character adjacent to it.

3.1.2 Pointing Cursor

The pointing cursor has the advantage of permitting the user to point at display information and select an item. This direct-manipulation approach presents the users with commands that do not have to be learned, reduces the chance of typographic errors on a keyboard, and keeps their attention on the display.

Use a cursor to enable users to move their focus of input or attention within a display. Make the pointing cursor available at all times. It should not obscure critical characters.



Succeed/Fail Counts

	B	C	D	E
1	Task	Succeed	Fail	Description
2				
3	1	6	3	SET ALARMS F
4	2	7	2	VERIFY THAT
5	3	8	1	CHANGE THE D
6				
7	1	9	1	SET ALARM FO
8	2	10	0	VERIFY THAT

Use crosshairs and outline tracing when fine positioning accuracy is required.

The placeholding cursor should "flash" to make it easier to locate; otherwise, it will be barely discernable from the rest of the text.

Make the Placeholding cursor the height or width of the character next to it.

Figure 3-1. Design of Cursors

3.1.2.1 Do not make the pointing cursor blink.

3.1.2.2 Make the pointing cursor completely graphic. Do not use a label.

3.1.2.3 Do not move the pointing cursor without input from the user.

3.1.2.3.1 Ensure that the step size of cursor movement is both horizontally and vertically consistent.

3.1.2.4 Use a cursor to enable the user to move the focal point of input or attention within a display.

3.1.2.5 Make the pointing cursor available at all times. It should not obscure other, critical information.

3.1.2.6 Use crosshairs when fine positioning accuracy is required.

3.1.2.7 Use multiple cursors only if they are required by the task.

3.1.2.7.1 Make multiple cursors visually distinctive.

3.1.2.7.2 Provide a visual indication of the cursor that is being controlled.

3.2 Text

The user interface presents text, for example, in menus, labels, help windows, and message areas. (Figure 3-2). The user may need to enter brief text, as in naming a file or making control entries. The user may need to enter continuous text, as in writing a comment or maintaining an on-line event log. The user needs simple editing capabilities (e.g., for correcting typographical errors and making word substitutions) that do not require going into a separate edit *mode*.

3.2.1 Text Fields

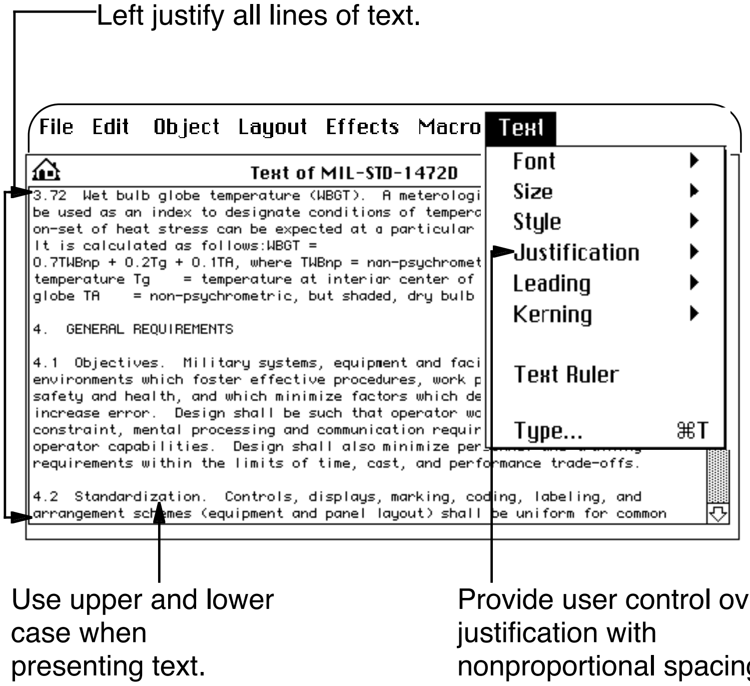
Present text fields in a consistent format, from one display to another.

3.2.1.1 Enable the use of text fields for the execution of commands, such as spell check, grammar check, search, find, and replace.

3.2.1.2 Ensure that control entries (e.g., keyed menu selections or commands) are distinguishable from displayed text to prevent the user from entering controls as text.

3.2.1.2.1 Permit users to specify units of text as (e.g., words, paragraphs) modifiers for control entries.

3.2.1.2.2 Highlight specified units of text to indicate boundaries of the text affected by control entries.


Left justify all lines of text.
File
Edit
Object
Layout
Effects
Macro
Text
Font
Size
Style
Justification
Leading
Kerning
Text Ruler
Type... ⌘T
Use upper and lower case when presenting text.
Provide user control over justification with nonproportional spacing.

Text of MIL-STD-1472D

```
3.72  Wet bulb globe temperature (WBGT).  A meterologi
be used as an index to designate conditions of tempera
on-set of heat stress can be expected at a particular
It is calculated as follows:WBGT =
0.7TWBnp + 0.2Tg + 0.1TA, where TWBnp = non-psychromet
temperature Tg    = temperature at interiar center of
globe TA    = non-psychrometric, but shaded, dry bulb

4.  GENERAL REQUIREMENTS

4.1  Objectives.  Military systems, equipment and faci
environments which foster effective procedures, work p
safety and health, and which minimize factors which de
increase error.  Design shall be such that operator wo
constraint, mental processing and communication requir
operator capabilities.  Design shall also minimize per
requirements within the limits of time, cast, and performance trade-offs.

4.2  Standardization.  Controls, displays, marking, coding, labeling, and
arrangement schemes (equipment and panel layout) shall be uniform for common
```

Figure 3-2. Presentation of Text

3.2.2 Continuous Text

Give users reasonable control over justification, line spacing, page structure (e.g., headers, margins, tab stops, and footers), and print options.

3.2.2.1 Left justify all lines of continuous text.

3.2.2.1.1 Do not permit right justification with non-proportional spacing.

3.2.2.2 Maintain constant spacing between words. Do not use proportional spacing.

3.2.2.3 Use 150% of character height as the default line spacing.

3.2.2.4 Display continuous text in upper and lower case letters (i.e., mixed case.)

3.2.2.5 Provide automatic line breaks for entry/editing of unformatted text.

3.2.2.5.1 Permit the user to override automatic line breaks by inserting page or section breaks when formatting and editing text.

3.2.2.6 For predefined page structures, provide the standard format automatically.

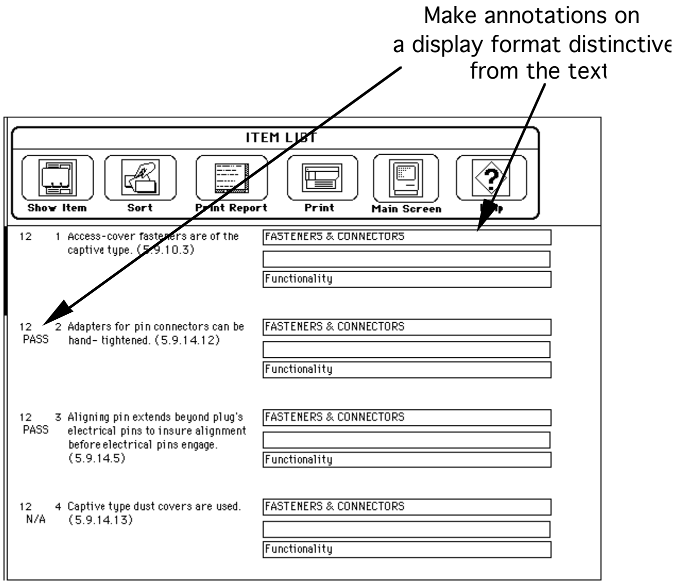
3.2.2.7 Allow users to label and store frequently used text formats and segments for future use.

3.2.2.8 Provide automatic pagination, but permit the user to override pagination in order to specify page numbers anywhere within a document.

3.2.2.8.1 Enable users to control for the number of lines in a paragraph that will be permitted to stand by themselves as "orphans" or "widows" at the top or bottom of a page.

3.2.2.9 Provide a capability for the display of annotations to displayed text.

3.2.2.9.1 Make annotations easy to distinguish from the text itself (Figure 3-3).



Make annotations on a display format distinctive from the text

12 1 Access-cover fasteners are of the captive type. (5.9.10.3)
FASTENERS & CONNECTORS
Functionality

12 PASS 2 Adapters for pin connectors can be hand- tightened. (5.9.14.12)
FASTENERS & CONNECTORS
Functionality

12 PASS 3 Aligning pin extends beyond plug's electrical pins to insure alignment before electrical pins engage. (5.9.14.5)
FASTENERS & CONNECTORS
Functionality

12 N/A 4 Captive type dust covers are used. (5.9.14.13)
FASTENERS & CONNECTORS
Functionality

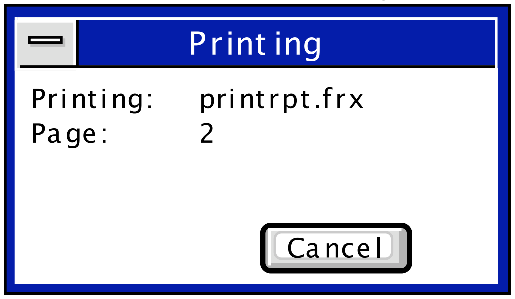
Figure 3-3. Annotations to Displayed Text

3.2.2.10 Permit the user to specify portions of the text for printing (e.g., a single page or range of pages).

3.2.2.10.1 Allow the user to display text as it will be printed, including underlining, boldface, subscript, superscript, special characters, special symbols, and different styles and sizes of type.

3.2.2.10.2 Inform the user of the status of requests for printouts (Figure 3-4). For example, notify the user when a printout has been completed.

Inform the user of printing status



Printing: printrpt.frx
Page: 2

Figure 3-4. Print Status Display

3.3 Fonts and Typography

The legibility of displayed text is a major UI issue. Reading from the screen can be considerably slower compared to reading from paper. The following guidelines assume adequate display contrast, depending on lighting conditions.

3.3.1 Font Size and Styles

Improve legibility by the use of well-formed letter shapes and type sizes that are appropriate for viewing distances (Figure□3-5).

3.3.1.1 For a nominal viewing distance of 18 inches, the minimum recommended size for a font is 0.08 inch (just slightly less than 1/16th of an inch).

3.3.1.2 Where alphanumeric characters are displayed, select font styles to allow discrimination of similar characters, such as letter l and number 1, letter Z and number 2.

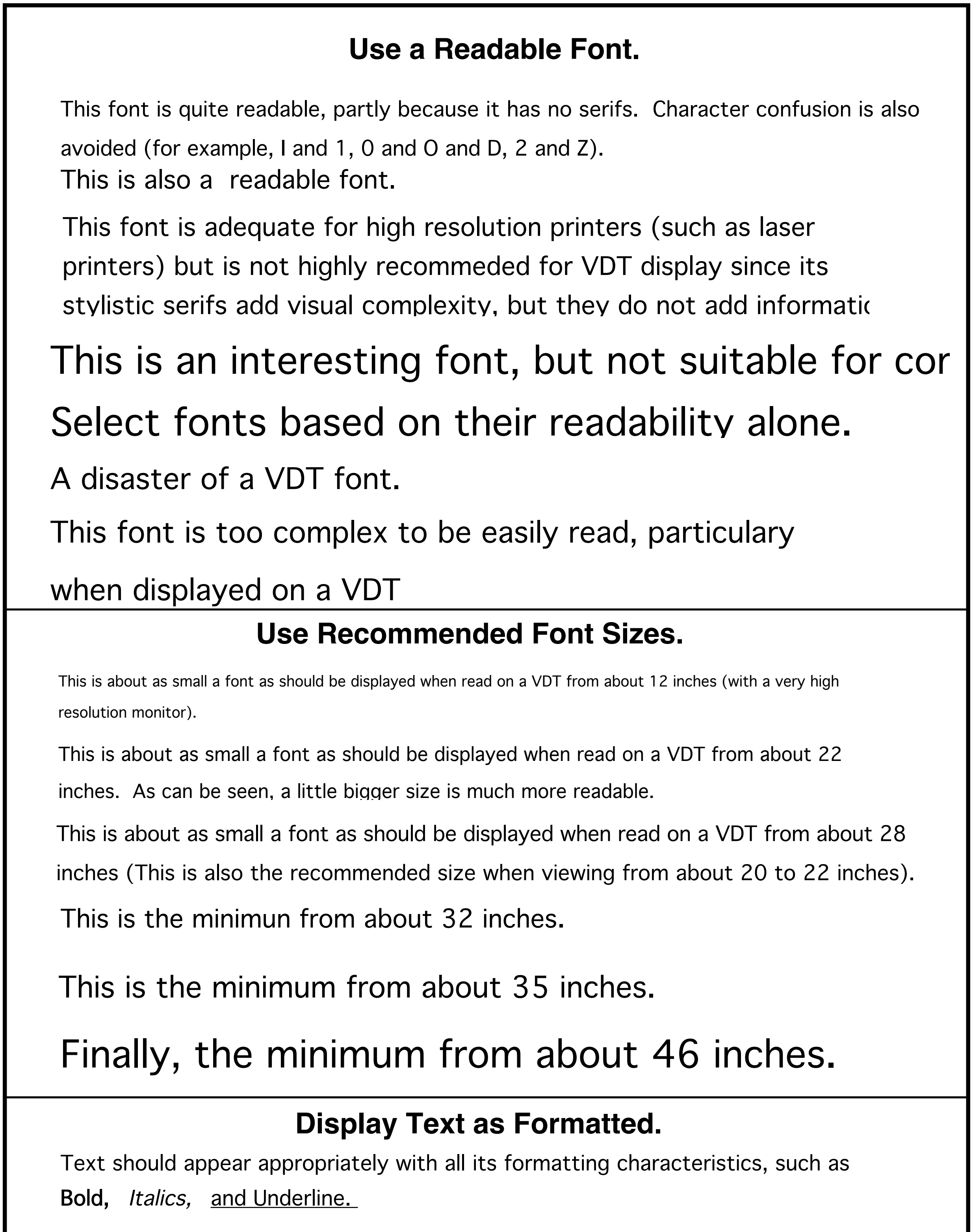
3.3.1.3 Do not use varying sizes or styles of fonts for any reason other than coding (for example, text as labels, text as data, text as command input).

3.3.1.4 Use selected fonts in a consistent fashion throughout the interface, and provide upper and lower case with full descenders.

3.3.1.5 Avoid type faces that have extended serifs, internal patterns, or that are striped, italicized, stenciled, shadowed or 3-dimensional, or fonts that appear like handwritten script or like Old English script, and fonts that are distorted to look tall and thin or wide and fat.

3.4 Tables

In general, users should not be required to search through lengthy tabular data to find required values. That is a job for the software. The following guidelines apply when there is no alternative to tabular presentation or when a brief table will support task performance.



Use a Readable Font.

This font is quite readable, partly because it has no serifs. Character confusion is also avoided (for example, I and 1, 0 and O and D, 2 and Z).

This is also a readable font.

This font is adequate for high resolution printers (such as laser printers) but is not highly recommeded for VDT display since its stylistic serifs add visual complexity, but they do not add informati

This is an interesting font, but not suitable for cor

Select fonts based on their readability alone.

A disaster of a VDT font.

This font is too complex to be easily read, particulary

when displayed on a VDT

Use Recommended Font Sizes.

This is about as small a font as should be displayed when read on a VDT from about 12 inches (with a very high resolution monitor).

This is about as small a font as should be displayed when read on a VDT from about 22 inches. As can be seen, a little bigger size is much more readable.

This is about as small a font as should be displayed when read on a VDT from about 28 inches (This is also the recommended size when viewing from about 20 to 22 inches).

This is the minimun from about 32 inches.

This is the minimum from about 35 inches.

Finally, the minimum from about 46 inches.

Display Text as Formatted.

Text should appear appropriately with all its formatting characteristics, such as **Bold,** *Italics,* and Underline.

Figure 3-5. Use of Font Sizes and Styles

3.4.1 Information Presented

Present information that compares detailed sets of data in a recognizable order. A consistent design is required to facilitate scanning and assimilation (Figure 3-6).

3.4.2 Headings

Use row and column headings that reflect information the user had before consulting the table.

3.4.2.1 For large tables exceeding one screen, provide column and row labels in all displayed sections of the table.

3.4.3 Arrangement

Arrange rows and columns according to some logic (e.g., chronological, alphabetic).

3.4.4 Scanning Cues

Provide adequate separation between columns (i.e., at least 3 spaces) and between groups of rows (e.g., a blank line inserted after every 5th row).

3.5 Lists

Display a series of related items in a list to support quick, accurate scanning (Figure 3-6).

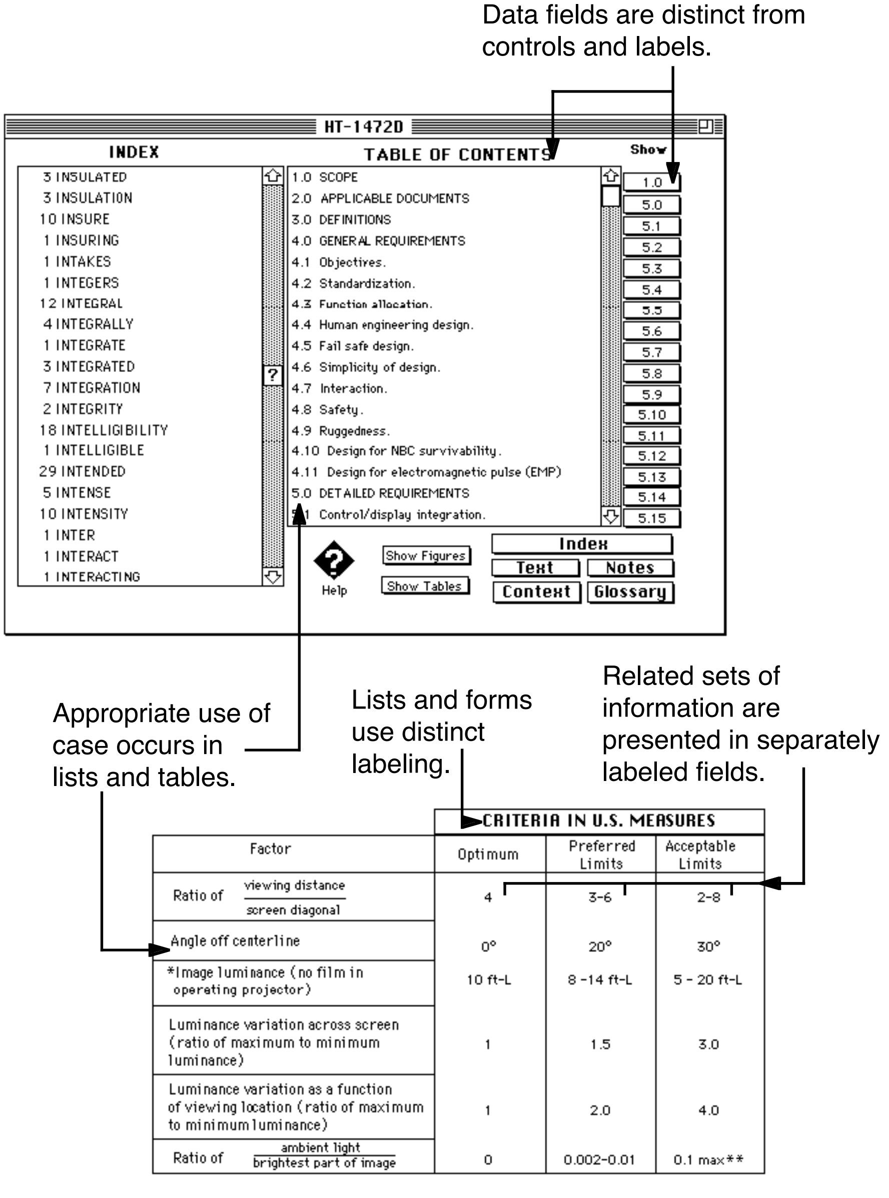
3.5.1 Number of Columns

A single column is usually recommended that is, each item in the list starts on a new line.

3.5.1.1 For a more compact display of a long list, use multiple columns. When using multiple columns, order items vertically within each column (Figure 3-6).

3.5.2 Order of Items

Maintain the same order of items for each instance of a particular list. Base the order of items on natural logic, such as frequency of use, related functionality, or the normal sequence of user actions. If there is no apparent logical basis for ordering items, list them alphabetically.



Data fields are distinct from controls and labels.

HT-1472D

INDEX

3 INSULATED
3 INSULATION
10 INSURE
1 INSURING
1 INTAKES
1 INTEGERS
12 INTEGRAL
4 INTEGRALLY
1 INTEGRATE
3 INTEGRATED
7 INTEGRATION
2 INTEGRITY
18 INTELLIGIBILITY
1 INTELLIGIBLE
29 INTENDED
5 INTENSE
10 INTENSITY
1 INTER
1 INTERACT
1 INTERACTING

TABLE OF CONTENTS

1.0 SCOPE
2.0 APPLICABLE DOCUMENTS
3.0 DEFINITIONS
4.0 GENERAL REQUIREMENTS
4.1 Objectives.
4.2 Standardization.
4.3 Function allocation.
4.4 Human engineering design.
4.5 Fail safe design.
4.6 Simplicity of design.
4.7 Interaction.
4.8 Safety.
4.9 Ruggedness.
4.10 Design for NBC survivability.
4.11 Design for electromagnetic pulse (EMP)
5.0 DETAILED REQUIREMENTS
5.1 Control/display integration.

Show

Appropriate use of case occurs in lists and tables.

Lists and forms use distinct labeling.

Related sets of information are presented in separately labeled fields.

	CRITERIA IN U.S. MEASURES		
Factor	Optimum	Preferred Limits	Acceptable Limits
Ratio of $\frac{\text{viewing distance}}{\text{screen diagonal}}$	4	3-6	2-8
Angle off centerline	0°	20°	30°
*Image luminance (no film in operating projector)	10 ft-L	8 -14 ft-L	5 - 20 ft-L
Luminance variation across screen (ratio of maximum to minimum luminance)	1	1.5	3.0
Luminance variation as a function of viewing location (ratio of maximum to minimum luminance)	1	2.0	4.0
Ratio of $\frac{\text{ambient light}}{\text{brightest part of image}}$	0	0.002-0.01	0.1 max**

Figure 3-6. Display of Tables and Lists

3.5.2.1 If a single item in a list continues on to another line, mark the item to indicate the continuation of the item, for example use an ellipsis (…).

3.5.2.2 If the list is very long, use a hierarchic structure to partition it into a set of more compact lists.

3.5.3 Orderly Format

Attention to alignment and labeling can improve UI consistency.

3.5.3.1 Align decimal points when listing numbers with decimal values.

3.5.3.2 When decimal values are not used, numbers are flushed right.

3.5.3.3 Alphabetic listings are flushed left.

3.5.3.4 Labels describe the contents of the lists and are flushed left or centered.

3.6 Pushbuttons

Maintain consistency in the design of pushbuttons. Although pushbuttons can vary in size and shape, the design of a particular type of pushbutton should remain consistent throughout an application. Standard names and uses for common pushbutton functions are listed in Table 3-1.

Table 3-1 Names and Uses for Pushbutton Functions

Names	Uses
OK	Confirms any information changed in a window and the window is closed.
Apply	Any changes in a window will occur and be displayed in the window.
Reset	Cancels any changes that have not been submitted.
Cancel	Closes a window after no changes were submitted.
Help	Displays contextual help for an item or help for an entire window.

3.6.1 Pushbutton Captions

Provide captions on pushbuttons, specifically stating the actions they are intended to perform.

3.6.2 Information Prior to Action

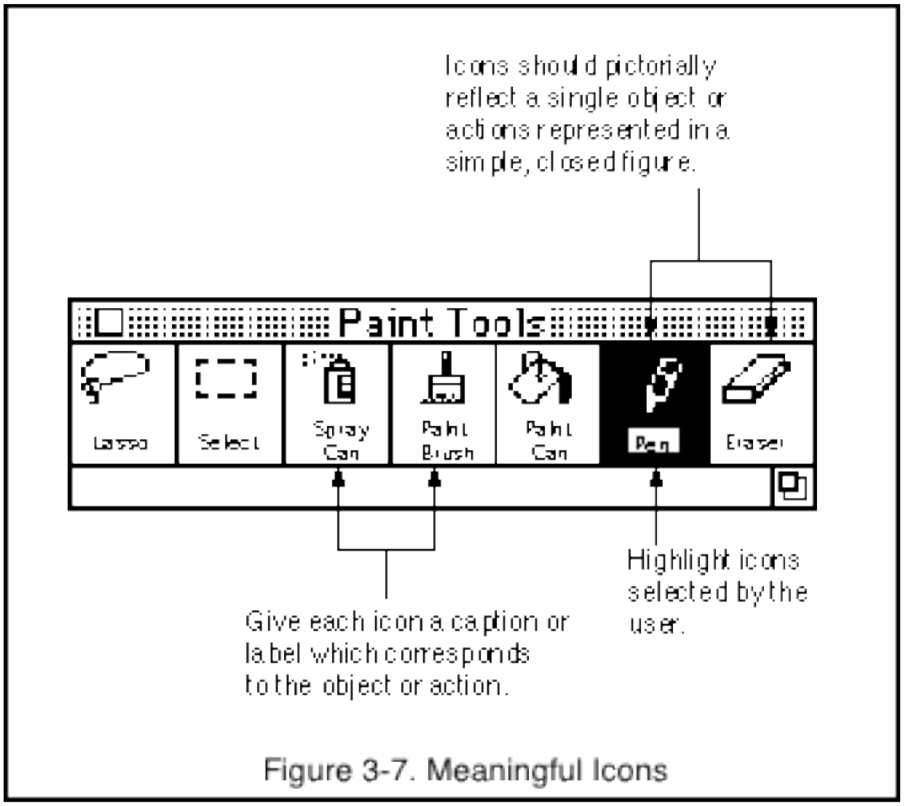
If additional information must be supplied by the user before the system can carry out a pushbutton action, provide an ellipsis (...) after the pushbutton caption to indicate that a query will be presented.

3.6.3 Arrangement of Pushbuttons

To the extent possible, arrange pushbuttons by frequency of use. For example, position frequently used action buttons to the left of or above other displayed buttons. When the same buttons are used for different windows, consistently place them in the same location, and keep related buttons together.

3.7 Icons

As illustrated in Figure 3-7, *icons* are pictorial representations of objects or actions that reflect the controlling metaphor of the application (e.g., desktop, office environment, artist's studio). Although it seems obvious that icons should look like what they represent, this is not always easy to achieve in practice. An icon that means one thing to a developer may mean something entirely different to a user. It is important to pre-test icons with users to verify that the icon set conveys the intended meanings.



Icons should pictorially reflect a single object or actions represented in a simple, closed figure.

Give each icon a caption or label which corresponds to the object or action.

Highlight icons selected by the user.

Figure 3-7. Meaningful Icons

3.7.1 Representation

Use icons that have clear meanings to their shape and pictorially reflect the objects or actions represented. To the extent possible, use icons that are already familiar to users.

3.7.1.1 Make each icon represent a single object or action.

3.7.1.2 Make each icon a simple, closed figure. Use as few graphical components as necessary. Avoid ornamentation.

3.7.1.3 Do not use purely abstract designs for icons.

3.7.1.4 Use the same icons for the same objects and actions across applications.

3.7.1.5 If it is necessary to create a new icon, consult standard symbol sets available from the American National Standards Institute (ANSI) and other sources to find established icons that may meet the need.

3.7.1.6 If no existing icons are satisfactory, create shapes that are meaningful to users, easily recognizable, and visually distinct from each other.

3.7.1.7 Use humor in icon design only if all users will get the joke and none will be confused or offended.

3.7.2 Size

Make the icon big enough to be seen, recognized, and selected easily.

3.3.2.1 Make all of the icons within a related set the same size.

3.3.2.2 Determine size requirments imposed by UI style guides, GUI software, and standards for legibility.

3.7.3 Number

For most applications, display fewer than twenty icons simultaneously on the same screen.

3.7.4 Labeling

Give each icon a text label corresponding to the object or action. Do not let the label obscure the icon.

3.7.4.1 Place the caption or label directly beneath the icon (Figure 3-7).

3.7.5 Grouping

Group related icons by using similar shapes and colors to depict a common relationship.

3.7.5.1 Provide a default arrangement, but permit users to rearrange icons within the context of their tasks.

3.7.6 Highlighting

Highlight icons selected by the user.

3.7.7 Documentation

Provide a glossary in on-line help containing a list of standard icons and their associated objects and actions.

3.7.8 Testing

Prior to implementation, measure the effectiveness of icons by testing them, with a representative group of users, for quick recognition and ease of learning. Re-design and re-test until usability objectives are met.

3.8 Labels

The quality of labeling employed throughout a user interface can impact user performance. Labels should be as meaningful and detailed as possible (Figure 3-8).

3.8.1 Wording

Give each data field, data group, message, pushbutton, icon, and window a descriptive label or caption, generally not a number.

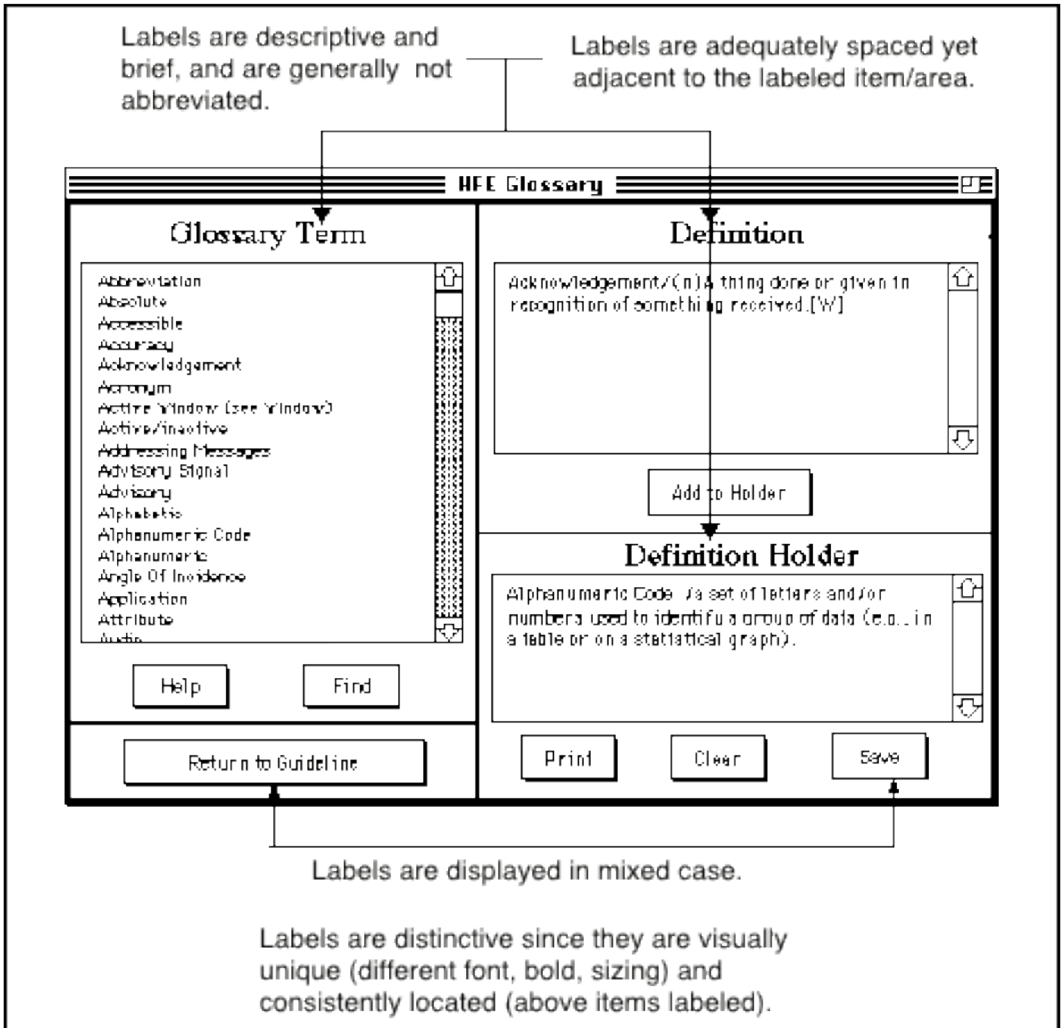
3.8.1.1 Spell words in full, using the simplest possible words as labels. Do not use contractions, abbreviations, or punctuation in labels (unless absolutely necessary for meaning or to accommodate space restrictions, or unless the label chosen is an accepted standard in the users' environment).

3.8.1.2 Display labels and titles in mixed case.

3.8.2 Location

Locate an item's label as close as possible to the item it describes (e.g., adjacent to the item, immediately above or below the item).

3.8.2.1 Integrate labels with graphics, using the same type style for labels and text.



Acknowledgement/(n)A thing done or given in recognition of something received.[W]

Alphanumeric Code /a set of letters and/or numbers used to identify a group of data (e.g., in a table or on a statistical graph).

Labels are descriptive and brief, and are generally not abbreviated.

Labels are adequately spaced yet adjacent to the labeled item/area.

Labels are displayed in mixed case.

Labels are distinctive since they are visually unique (different font, bold, sizing) and consistently located (above items labeled).

Figure 3-8. High-Quality Labeling

3.8.3 Orientation

Display labels in a left-to-right (horizontal) orientation. Do not display labels vertically or in any other off-horizontal orientation.

3.8.4 Differentiation

Differentiate labels from other screen elements in a unique and consistent manner (e.g., bold, underlined).

3.8.5 Spacing

Separate labels from one another by at least two standard character spaces.

3.8.6 Consistency

Make labels consistent in wording, location, orientation, differentiation, and spacing throughout an application or set of applications.

3.9 Check Boxes and Radio Buttons

Use *check boxes* or *radio buttons* to present multiple options. Multiple options are often presented in panels of check boxes or radio buttons, but their use is not interchangeable. Check boxes are used to allow the user to select several options. Radio buttons permit only one selection from a group of options.

3.9.1 Labeling

Descriptions of alternative choices for check boxes and radio buttons should be fully and clearly spelled out and positioned to the right of each box or button.

3.9.1.1 Display descriptors (labels) in mixed case.

3.9.1.2 Capitalize the first letter of each major word in multiword labels.

3.9.2 Choice Indication

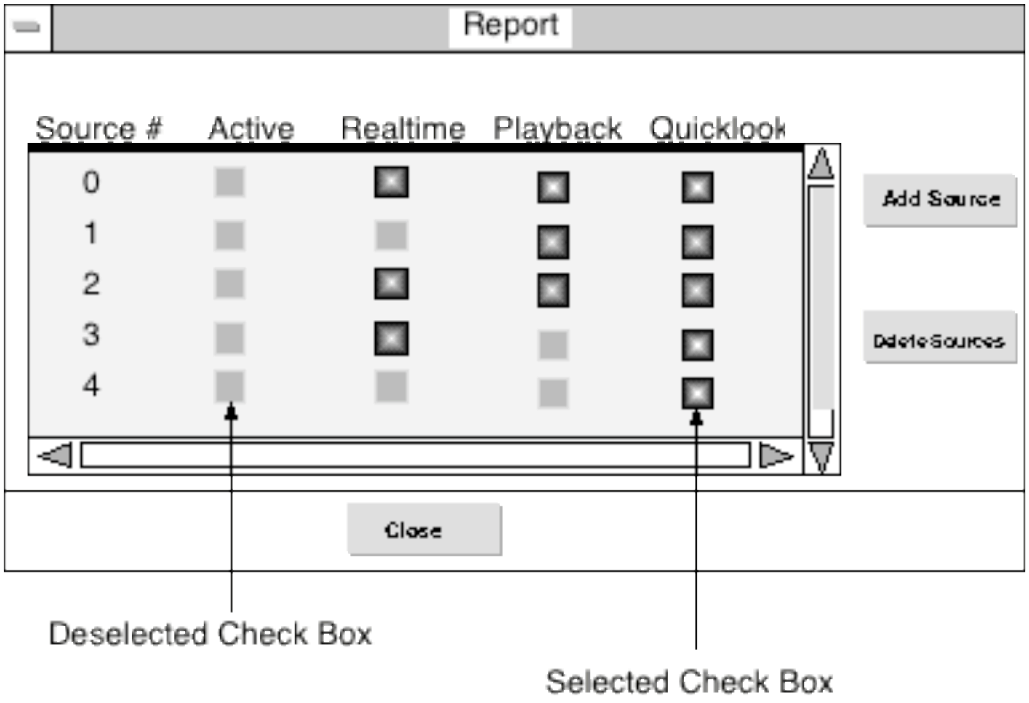
Indicate that a choice has been made, in the box or button, by means of a check mark, fill-in, or highlighting.

3.9.3 Check Boxes

Use check boxes for non-exclusive options. When options are not mutually exclusive, check boxes allow for more than one selection to be made. Check boxes are usually displayed as square or rectangular boxes with option labels inscribed alongside each box (Figure 3-9).

3.9.3.1 Provide labels for each set of check boxes.

3.9.3.2 Arrange check boxes in logical order so that the most frequently used boxes are at the top or at the left, depending on how the boxes are oriented.



Source #	Active	Realtime	Playback	Quicklook
0	□	☒	☒	☒
1	□	□	☒	☒
2	□	☒	☒	☒
3	□	☒	□	☒
4	□	□	□	☒

Deselected Check Box

Selected Check Box

Figure 3-9. Examples of Check Boxes

3.9.3.3 A columnar orientation is generally preferred for check boxes, with the boxes aligned to the left.

3.9.3.3.1 If there is limited space, a horizontal orientation may be used, with adequate separation (three spaces) between each box.

3.9.3.4 Within a group of checkboxes, make the boxes equal in height and width.

3.9.3.5 Label style and orientation should remain consistent to the extent possible.

3.9.3.6 Use the method of indicating selected options consistently across all panels of check boxes in an application or set of applications.

3.9.4 Radio Buttons

Use circular or diamond-shaped radio buttons for mutually exclusive options (Figure 3-10). Avoid using retangular radio buttons because they may be confused with check boxes. When options are mutually exclusive, radio buttons permit the selection of only one item at a time.

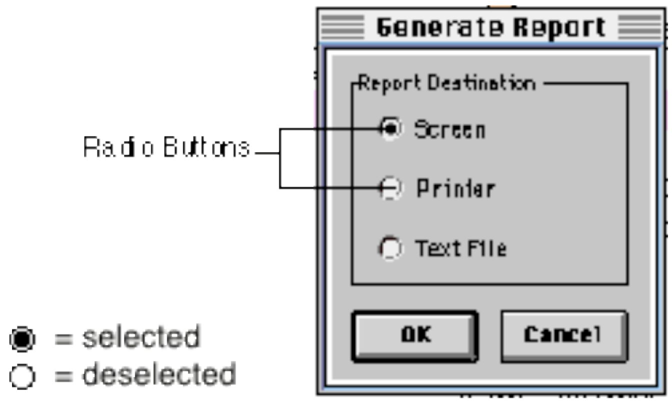


Figure 3-10. Examples of Radio Buttons

3.9.4.1 Provide labels for each set of radio buttons.

3.9.4.1.1 If a screen or window contains only one panel of radio buttons, the screen or window title may serve as the panel label (if there is no question of user confusion).

3.9.4.2 Limit to eight the number of options presented in a panel of radio buttons.

3.9.4.2.1 When nine or more options must be presented, consider using a scrollable list or a drop-down list.

3.9.4.3 Arrange options in a logical or expected order, beginning at the top of the panel.

3.9.4.4 Left-align the radio buttons and their labels in the preferred columnar format, as illustrated in Figure 3-10.

3.9.4.4.1 If the buttons must be arranged horizontally, provide at least three spaces between an option label and the next button.

3.9.4.5 When a particular option is not available, display it as subdued or grayed-out in relation to the brightness of the available options.

3.9.4.6 Make the selection target area include the radio button and its label.

3.9.4.6.1 When the cursor has been moved to an option, making it available for selection, highlight only the label (not the button) using a technique such as reverse video, reverse color, or a dashed box around the label.

3.9.4.7 Use the method chosen to indicate selection consistently across all panels of radio buttons in the application or set of applications.

3.10 Dialog Boxes

Dialog boxes ask users to specify preferences and to acknowledge messages from the system.

3.10.1 Dialog-Box Basics

There are two basic categories of dialog boxes, modal and modeless (Figure 3-11). Modal dialog boxes require a response before any further action can be taken by the user. Modeless dialog boxes allow the user to perform parallel dialogs. Switching between the modeless dialog box and its associated window is permitted. Palettes and toolbars are modeless dialog boxes.

3.10.1.1 Give each dialog box a title.

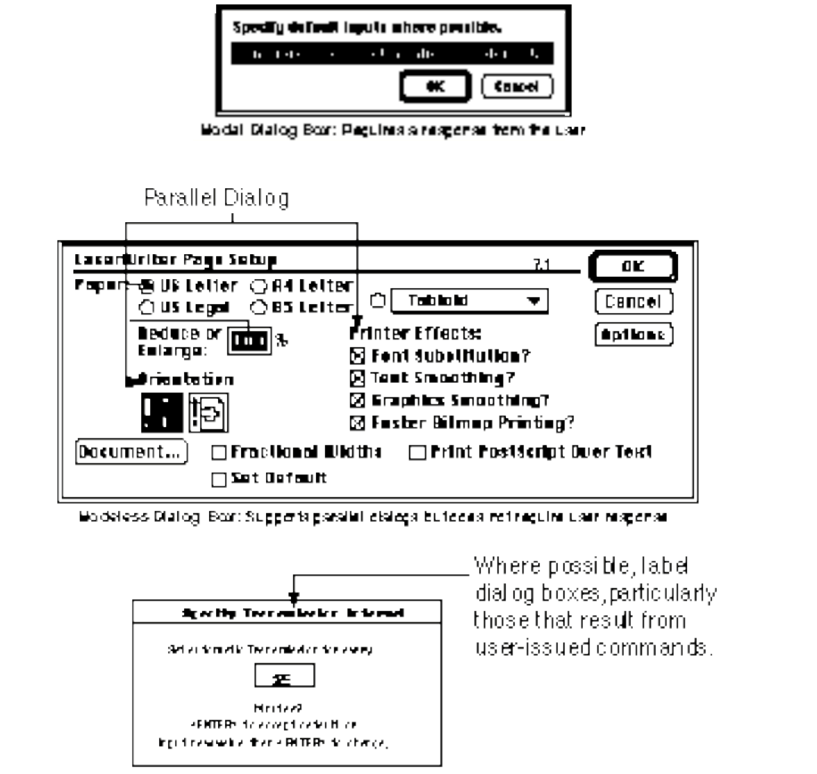
3.10.1.2 Match pushbuttons to the function of the dialog box.

3.10.1.3 Size dialog boxes to be smaller than application windows.

3.10.1.4 Locate each dialog box uniquely, depending on the scope of its relationship to system or application elements.

3.10.2 Message Boxes

When a warning of an unexpected event (e.g., printer out of paper) or information regarding an irreversible state is presented to the user (Figure 3-12), a message box provides space for user acknowledgment.



Specify default inputs where possible.

Modal Dialog Box: Requires a response from the User

LaserWriter Page Setup 7.1

Paper: ● US Letter ○ A4 Letter ○ US Legal ○ B5 Letter ○ Tabloid

Reduce or Enlarge: 100 %

Orientation

Printer Effects:
☒ Font Substitution?
☒ Text Smoothing?
☒ Graphics Smoothing?
☒ Faster Bitmap Printing?

□ Fractional Widths □ Print PostScript Over Text
□ Set Default

Modeless Dialog Box: Supports parallel dialogs but does not require User response

Specify Temperature Interval

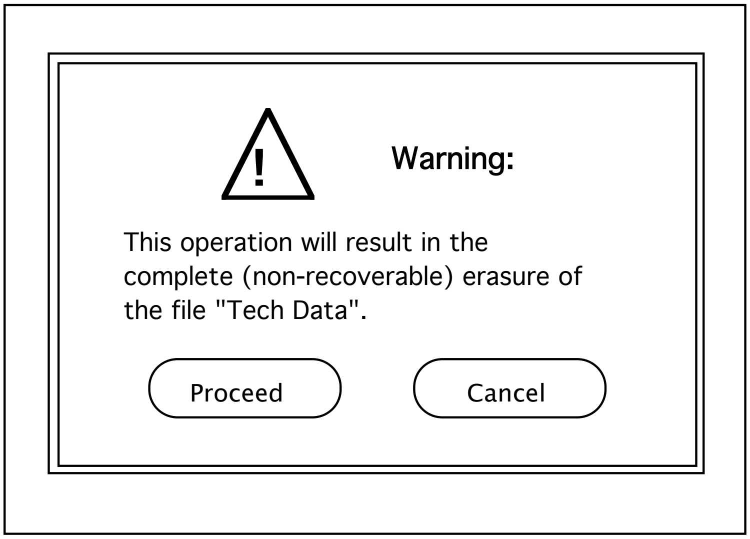
Select desired Temperature Interval

25

Minutes?
<ENTER> to accept default or
Input new value then <ENTER> to change.

Where possible, label dialog boxes, particularly those that result from user-issued commands.

Figure 3-11. Dialog Boxes



Warning:

This operation will result in the complete (non-recoverable) erasure of the file "Tech Data".

Figure 3-12. Message Box

References

3.0	Carlow (1992); Galitz 1993); Mayhew (1992); Shneiderman (1992); Smith and Mosier (1986)
3.1	Carlow (1992)
3.1.1	Carlow (1992)
3.1.1.1	Carlow (1992)
3.1.1.2	Carlow (1992)
3.1.1.3	Carlow (1992)
3.1.1.4	Carlow (1992)
3.1.2	Carlow (1992); Shneiderman (1992)
3.1.2.1	Carlow (1992)
3.1.2.2	Carlow (1992)
3.1.2.3	Carlow (1992); Smith and Mosier (1986)
3.1.2.3.1	Smith and Mosier (1986)
3.1.2.4	Carlow (1992)
3.1.2.5	Carlow (1992); Smith and Mosier (1986)
3.1.2.6	Carlow (1992)
3.1.2.7	Carlow (1992); Smith and Mosier (1986)
3.1.2.7.1	Carlow (1992); Smith and Mosier (1986)
3.1.2.7.2	Smith and Mosier (1986)
3.2	Smith and Mosier (1986)
3.2.1	Smith and Mosier (1986)
3.2.1.1	Galitz (1993); Smith and Mosier (1986)
3.2.1.2	Smith and Mosier (1986)
3.2.1.2.1	Smith and Mosier (1986)
3.2.1.2.2	Smith and Mosier (1986)
3.2.2	Carlow (1992)
3.2.2.1	Carlow (1992)
3.2.2.1.1	Carlow (1992)
3.2.2.2	Smith and Mosier (1986)
3.2.2.3	Carlow (1992)
3.2.2.4	Carlow (1992)
3.2.2.5	Smith and Mosier (1986)
3.2.2.5.1	Smith and Mosier (1986)
3.2.2.6	Smith and Mosier (1986)
3.2.2.7	Smith and Mosier (1986)
3.2.2.8	Smith and Mosier (1986)
3.2.2.8.1	Smith and Mosier (1986)
3.2.2.9	Smith and Mosier (1986)
3.2.2.9.1	Carlow (1992); Smith and Mosier (1986)
3.2.2.10	Smith and Mosier (1986)
3.2.2.10.1	Smith and Mosier (1986)
3.2.2.10.2	Smith and Mosier (1986)
3.3	Gould and Grischkowsky (1984); Horton (1991); Shneiderman (1992); Smith and Mosier (1986)

3.3.1	Horton (1991)
3.3.1.1	Carlow (1992)
3.3.1.2	Carlow (1992)
3.3.1.3	Carlow (1992)
3.3.1.4	Carlow (1992)
3.3.1.5	Carlow (1992)
3.4	Smith and Mosier (1986)
3.4.1	Smith and Mosier (1986)
3.4.2	Smith and Mosier (1986)
3.4.2.1	Smith and Mosier (1986)
3.4.3	Smith and Mosier (1986)
3.4.4	Smith and Mosier (1986)
3.5	Smith and Mosier (1986)
3.5.1	Smith and Mosier (1986)
3.5.1.1	Smith and Mosier (1986)
3.5.2	Smith and Mosier (1986)
3.5.2.1	Smith and Mosier (1986)
3.5.2.2	Smith and Mosier (1986)
3.5.3	Smith and Mosier (1986)
3.5.3.1	Smith and Mosier (1986)
3.5.3.2	Smith and Mosier (1986)
3.5.3.3	Smith and Mosier (1986)
3.5.3.4	Smith and Mosier (1986)
3.6	Galitz (1993)
3.6.1	Galitz (1993)
3.6.2	Galitz (1993)
3.6.3	Galitz (1993)
3.7	Galitz (1993); Horton (1994); Mayhew (1992); Shneiderman (1992)
3.7.1	Carlow (1992); Galitz (1993); Horton (1994)
3.7.1.1	Carlow (1992); Galitz (1993); Horton (1994)
3.7.1.2	Carlow (1992); Galitz (1993); Horton (1994); Mayhew (1992)
3.7.1.3	Carlow (1992); Horton (1994)
3.7.1.4	Carlow (1992); Horton (1994); Mayhew (1992)
3.7.1.5	American National Standards Institute, 11 West 42nd Street, New York, NY 10036 (voice: 212-642-4900; Fax: 212-302-1286); Horton (1994, Icon Starter Set Disk); Fowler and Stanwick (1995); Mayhew (1992)
3.7.1.6	Horton (1994); Mayhew (1992)
3.7.1.7	Carlow (1992); Horton (1994)
3.7.2	Carlow (1992); Horton (1994)
3.7.2.1	Horton (1994)
3.7.2.2	Horton (1994)
3.7.3	Galitz (1993); Mayhew (1992)
3.7.4	Carlow (1992); Horton (1994)
3.7.4.1	Carlow (1992)
3.7.5	Galitz (1992); Horton (1994)
3.7.5.1	Galitz (1993); Mayhew (1992)

3.7.6	Carlow (1992); Horton (1994)
3.7.7	Carlow (1992)
3.7.8	Galitz (1993); Horton (1994)
3.8	Galitz (1993)
3.8.1	Galitz (1993); Carlow (1992)
3.8.1.1	Galitz (1993); Carlow (1992)
3.8.1.2	Code 520 preference
3.8.2	Carlow (1992)
3.8.2.1	Galitz (1993)
3.8.3	Galitz (1993)
3.8.4	Carlow (1992); Smith and Mosier (1986)
3.8.5	Carlow (1992)
3.8.6	Carlow (1992); Smith and Mosier (1986)
3.9	Fowler and Stanwick (1995); Galitz (1993)
3.9.1	Galitz (1993)
3.9.1.1	Galitz (1993)
3.9.1.2	Galitz (1993)
3.9.2	Galitz (1993)
3.9.3	Galitz (1993)
3.9.3.1	Galitz (1993)
3.9.3.2	Galitz (1993)
3.9.3.3	Galitz (1993)
3.9.3.3.1	Galitz (1993)
3.9.3.4	Galitz (1993)
3.9.3.5	Galitz (1993)
3.9.3.6	Galitz (1993)
3.9.4	Fowler and Stanwick (1995); Galitz (1993) Code 520 preference for non-rectangular shape
3.9.4.1	Galitz (1993)
3.9.4.1.1	Galitz (1993)
3.9.4.2	Galitz (1993)
3.9.4.2.1	Galitz (1993)
3.9.4.3	Galitz (1993)
3.9.4.4	Galitz (1993)
3.9.4.4.1	Galitz (1993)
3.9.4.5	Galitz (1993)
3.9.4.6	Galitz (1993)
3.9.4.6.1	Galitz (1993)
3.9.4.7	Galitz (1993)
3.10	Fowler and Stanwick (1995)
3.10.1	Fowler and Stanwick (1995)
3.10.1.1	Fowler and Stanwick (1995)
3.10.1.2	Fowler and Stanwick (1995)
3.10.1.3	Fowler and Stanwick (1995)
3.10.1.4	Fowler and Stanwick (1995)
3.10.2	Carlow (1992); Galitz (1993)

4.0 Guidelines for Screen Layout and Design

This section is concerned with orderly design to support the exchange of information between the user and the computer. Of interest are screen format and content, such as where information is placed, how information is structured, and what information is included. Achieving consistency in screen design throughout an application's UI is a key challenge to the development team.

4.1 General Layout

The objective is to achieve a screen layout that meets users' information requirements while appearing well organized and uncluttered (Figure 4-1).

4.1.1 Presentation of Information

Present information in a directly usable form; the user should not have to cross-reference other resources or perform mental transformations of data.

4.1.1.1 To the extent possible, put all the data related to one task on a single screen. Users should be able to see the whole page on which they are working.

4.1.1.1.1 If all the required data will not fit on a single screen, divide it up into logical units so that the data needed first are presented first, followed by a screen or screens containing the data needed later in the task.

4.1.1.1.1.1 Indicate display continuation when display output is presented on more than one screen.

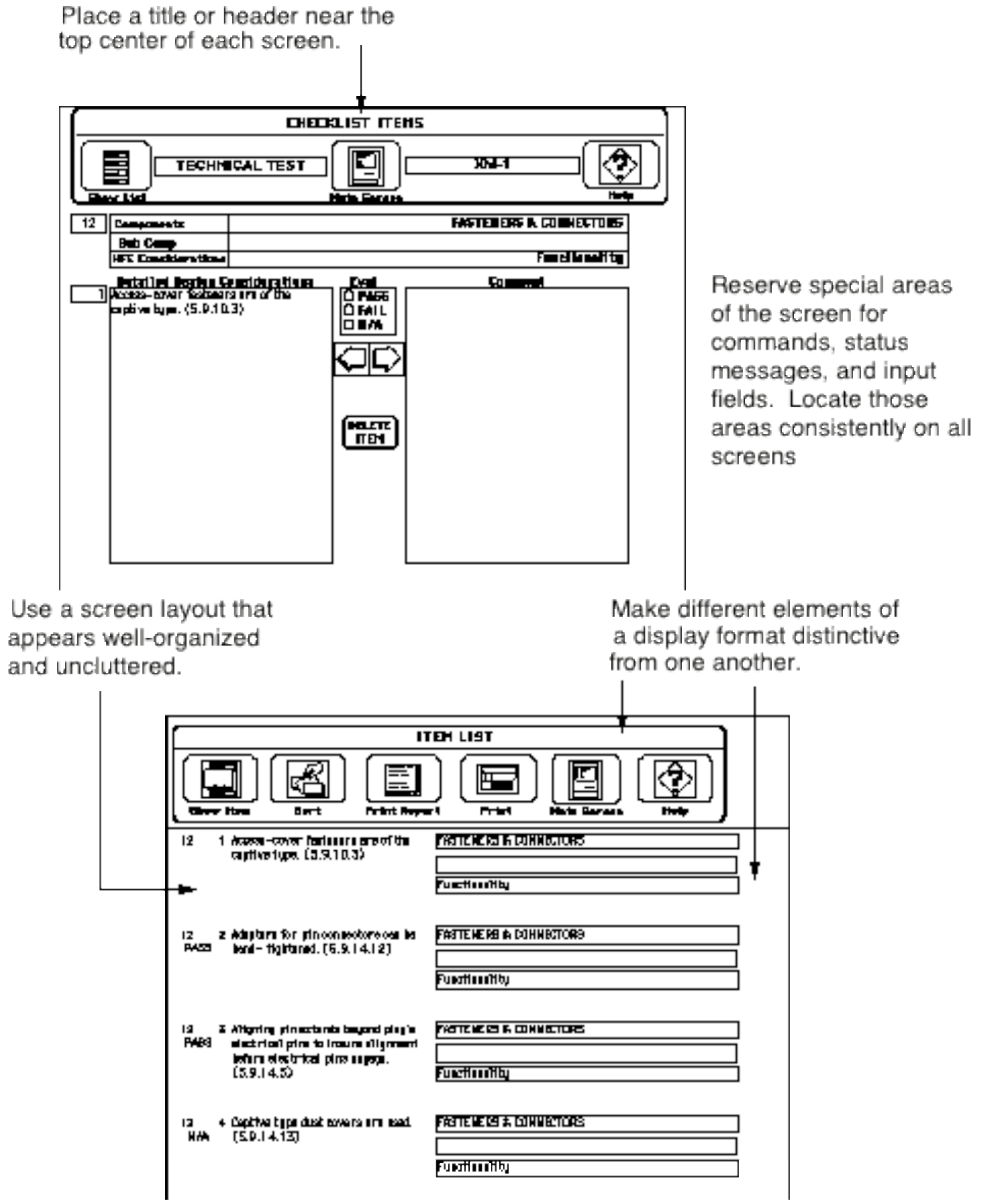
4.1.1.1.1.2 Label each screen of a multi-screen display to show its relationship to the others.

4.1.1.1.2 Display functionally related data items on the same screen. Do not require the user to jump back and forth within a set of screens or windows to find required pieces of data.

4.1.1.2 Reserve special areas of the screen for:

- Commands
- Status messages
- Input fields

Locate these areas consistently on all screens.



12 Component: FASTENERS & CONNECTORS

Sub Comp

HFE Considerations Functionality

Detailed Design Considerations | Eval | Comment

1 Access-cover fasteners are of the captive type. (5.9.10.3)

□ PASS
□ FAIL
□ N/A

12 1 Access-cover fasteners are of the captive type. (5.9.10.3) FASTENERS & CONNECTORS Functionality

12 PASS 2 Adapters for pin connectors can be hand-tightened. (5.9.14.12) FASTENERS & CONNECTORS Functionality

12 PASS 3 Aligning pins extend beyond plug's electrical pins to insure alignment before electrical pins engage. (5.9.14.5) FASTENERS & CONNECTORS Functionality

12 N/A 4 Captive type dust covers are used. (5.9.14.13) FASTENERS & CONNECTORS Functionality

Place a title or header near the top center of each screen.

Reserve special areas of the screen for commands, status messages, and input fields. Locate those areas consistently on all screens

Use a screen layout that appears well-organized and uncluttered.

Make different elements of a display format distinctive from one another.

Figure 4-1. Arranging Data on the Screen

4.1.1.3 To the extent possible, do not require users to perform mental comparisons or other analyses of data. Present the information integrated into the form needed for task performance (e.g., trends over time).

4.1.1.3.1 When there is no alternative to requiring users to detect similarities, differences, trends, and relationships across sets of data, format the screen so that the data are grouped to facilitate analysis and comparison.

4.1.2 Emphasis between Elements

Use contrasting display features to emphasize differences between elements:

- Different screen components, such as dialog boxes, windows, menus.
- Items being acted upon.
- Urgent items.

4.1.3 Visual Guidance through Grouping

Guide users through the screen with bordering lines formed by display elements, and provide symmetry and balance through the use of white space.

4.1.3.1 Group data in ways that support the users' logical train of thought in performing their tasks (Figure 4-2).

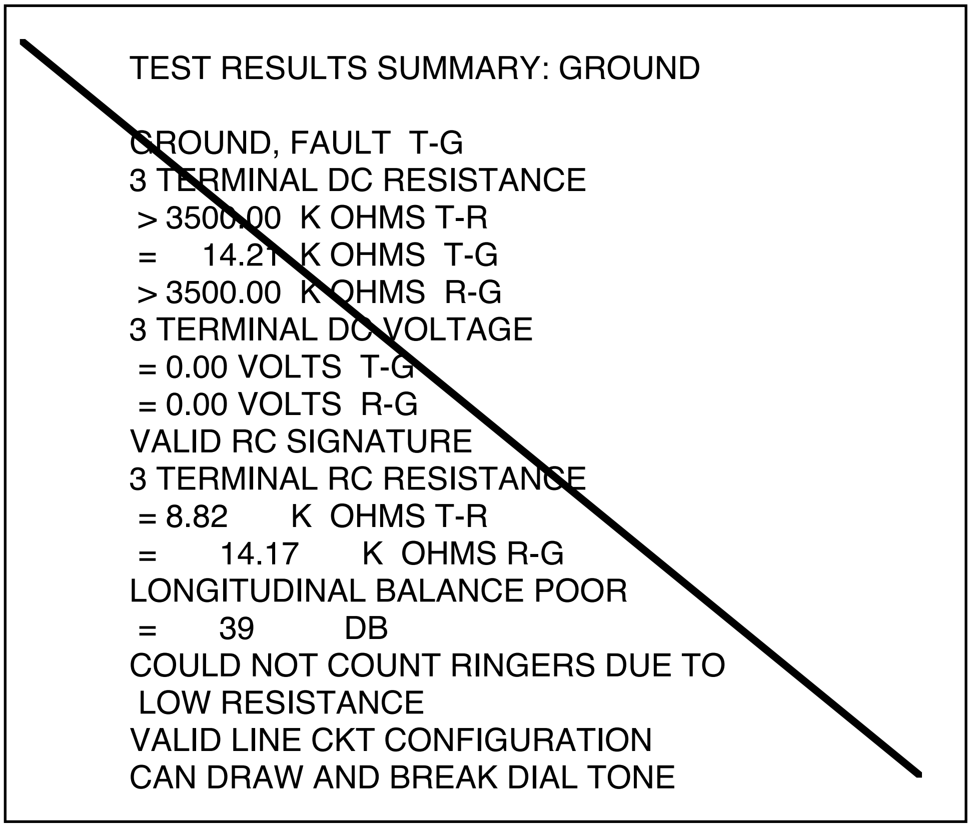
4.1.3.1.1 Group particularly important data items, and display them at the top of the screen.

4.1.3.1.2 Group data items that are used more frequently than others, and display them near the top of the screen (below the important items referred to in 4.1.3.1.1).

4.1.3.1.3 When data are used in some spatial or temporal order, group by sequence of use.

4.1.3.1.4 When sets of data are associated with particular questions or are related to particular functions, group each set to illustrate those functional relationships.

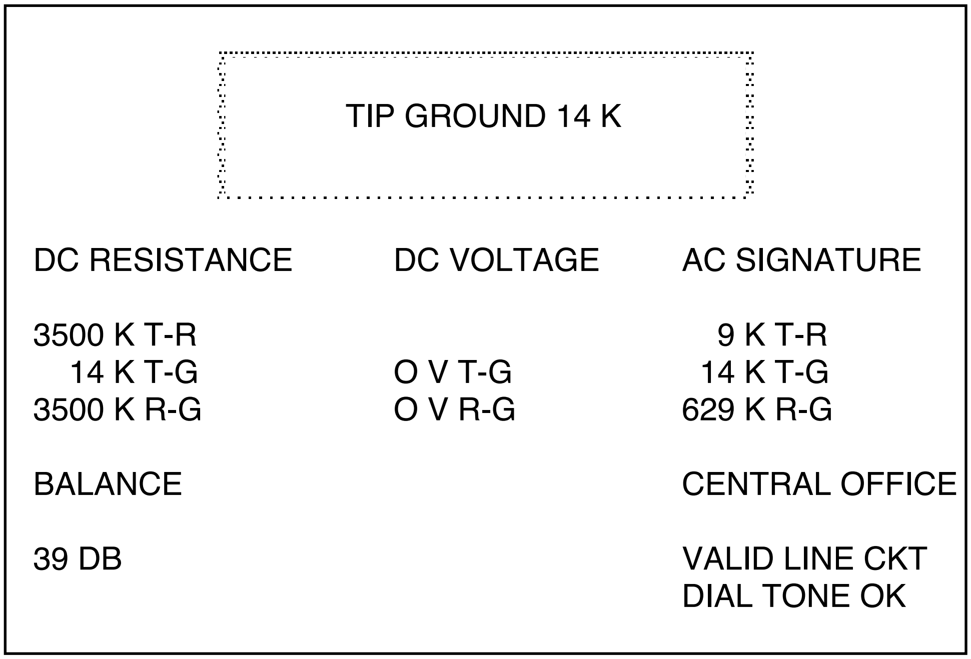
4.1.3.1.5 Use general-to-specific grouping when there are hierarchical relationships among data elements, with general elements preceding specific elements.



```
TEST RESULTS SUMMARY: GROUND

GROUND, FAULT  T-G
3 TERMINAL DC RESISTANCE
 > 3500.00  K OHMS T-R
 =     14.21  K OHMS  T-G
 > 3500.00  K OHMS  R-G
3 TERMINAL DC VOLTAGE
 = 0.00 VOLTS  T-G
 = 0.00 VOLTS  R-G
VALID RC SIGNATURE
3 TERMINAL RC RESISTANCE
 = 8.82       K  OHMS T-R
 =       14.17       K  OHMS R-G
LONGITUDINAL BALANCE POOR
 =       39          DB
COULD NOT COUNT RINGERS DUE TO
 LOW RESISTANCE
VALID LINE CKT CONFIGURATION
CAN DRAW AND BREAK DIAL TONE
```

Before: Ungrouped Data



```
                    TIP GROUND 14 K

DC RESISTANCE         DC VOLTAGE       AC SIGNATURE

3500 K T-R                                9 K T-R
  14 K T-G            O V T-G            14 K T-G
3500 K R-G            O V R-G           629 K R-G

BALANCE                                CENTRAL OFFICE

39 DB                                  VALID LINE CKT
                                       DIAL TONE OK
```

After: Data Selected and Grouped to Support the User's Task

Figure 4-2. Ease of Visual Detection through Grouping

(Source: Tullis, 1981)

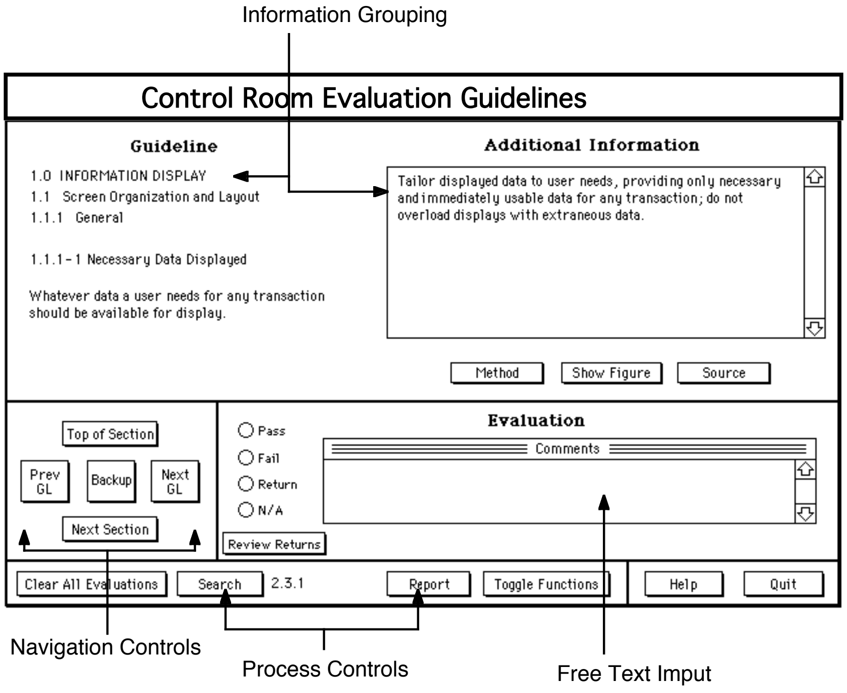
4.1.3.1.6 If there is no other known logic for grouping data, based on the cognitive task analysis, use a standard grouping technique (e.g., chronological, alphabetical).

4.1.3.2 Use grouping to make the screen appear as an organized collection of smaller identifiable elements.

4.1.3.2.1 Demarcate groups of information by spacing, drawing lines, color coding, or some other means (Figure 4-3). Be careful not to clutter the display with too many lines or colors.

4.1.3.2.1.1 For critical tasks, use at least one character space above and below, and two character spaces before and after the critical information.

Enhancement of Grouping through Borders



Control Room Evaluation Guidelines

Guideline

1.0 INFORMATION DISPLAY
1.1 Screen Organization and Layout
1.1.1 General

1.1.1-1 Necessary Data Displayed

Whatever data a user needs for any transaction should be available for display.

Additional Information

Tailor displayed data to user needs, providing only necessary and immediately usable data for any transaction; do not overload displays with extraneous data.

Evaluation

○ Pass
○ Fail
○ Return
○ N/A

Figure 4-3. Bordered Groupings

4.1.4 Visual Appearance

Be consistent with visual appearance and procedural usage.

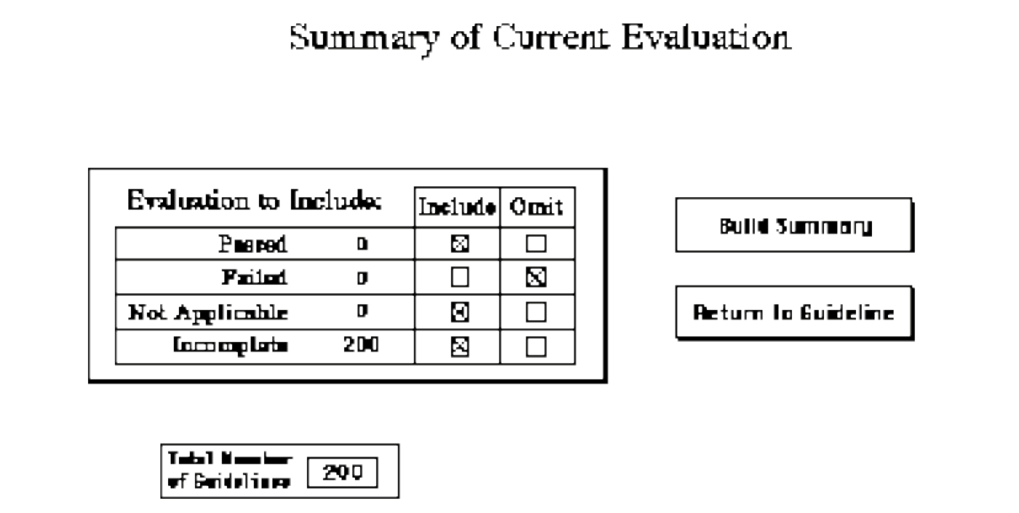
4.1.4.1 Place recurring data fields in consistent relative positions within displays.

4.1.4.2 When appropriate for users, use the same format for input and output. (Determine appropriateness through the task analysis and discussions with users.)

4.1.4.3 When appropriate, match data-entry formats and source-document formats. (Determine appropriateness through the task analysis and discussions with users.)

4.1.4.4 In general, keep *screen density* below 50% and preferably less than 25% (Figure□4-4).

4.1.4.4.1 In displays used for critical tasks, minimize screen density (the ratio of filled to unfilled pixels).



Summary of Current Evaluation

Evaluation to Include:		Include	Omit
Passed	0	☒	□
Failed	0	□	☒
Not Applicable	0	☒	□
Incomplete	200	☒	□

Total Number of Guidelines 200

Figure 4-4. Example Screen with Adequate Screen Density

4.1.5 Output Displays

Based on results of the ask analysis, provide the user with whatever data are needed for any transaction, making such data available for display. Use task analysis to determine detailed information requirements for the user. Users should not have to recall data from one *output display* to the next.

4.1.5.1 Continue to provide prior data when necessary so that users do not have to remember previous values when they are interpreting new data.

4.1.5.2 Use familiar, task-oriented terms on displayed data and labels. Ensure that carefully chosen words and the same grammatical structure are used consistently within and across displays.

4.1.5.3 Use complete words instead of abbreviations, whenever possible.

4.1.5.3.1 When using abbreviations, choose those that are most commonly used and familiar to the user.

4.1.5.3.2 Do not use the same abbreviation for different words; make sure abbreviations are distinctive.

4.1.5.3.3 Minimize punctuation of abbreviations and acronyms, for example, display USA instead of U.S.A.

4.1.5.3.4 Use a standard method to form abbreviations (Table 4-1).

4.1.5.3.5 Special abbreviations are used only when they are required for clarity. If an abbreviation must deviate from the standard method for forming abbreviations, make the deviation minimal. Keep abbreviations as short as possible.

4.1.5.3.6 When abbreviations or acronyms are required, provide a dictionary of terms in on-line help and in the user's guide.

4.1.5.4 Define the frequency of displaying a data field as required or optional. A required field is always displayed whenever a screen is used. An optional field is displayed when information is requested but not necessarily required. Alternatively, display both required and optional fields, but code them differently (e.g., place a parenthesis around the label of an optional field).

4.1.6 Entry Screens

If forms of any type are used for data entry, they should be compatible with those used for data output. Use the same item labeling and ordering for both. Data should be entered in units that are familiar to the user.

4.1.6.1 Ensure that screen titles reflect the names of the data elements. Clarification of titles on system worksheets may be required.

4.1.6.2 Minimize the amount of data that the user must enter.

4.1.6.2.1 The user should not have to enter the same data more than once. Correct items are saved and there is no need to enter those data again while changing incorrect items.

4.1.6.2.2 Preserve the context of each data-entry transaction.

4.1.6.3 During data entry, display feedback for each keyed entry within 0.02 seconds.

4.1.6.3.1 Design data-entry transactions and associated displays so that users are able to remain with one method of entry. If data entry is assigned to specific areas, provide clear visual definition of the entry fields as shown in Figure 4-5.

Table 4-1. Rules for Forming Abbreviations

Words and Word Groups in General

Single Word:

- Determine number of characters requirec
- Remove suffixes
- Choose first letter and last consonar
- Fill remaining space with consonants in the order in which they appea
- If insufficient consonants, use first vowe

Two-Word Group:

- Determine number of characters requirec
- Use single word methodology to take half of the characters from first word and half from the secon
- If an odd number of total characters is needed, take the extra charac from the longer worc

Three or More Words

- Form an acronym by using the first letter of each wo
- If additional letters are required, take them from the last word in tl manner of single word abbreviatinc

Commands

Multiple Words:

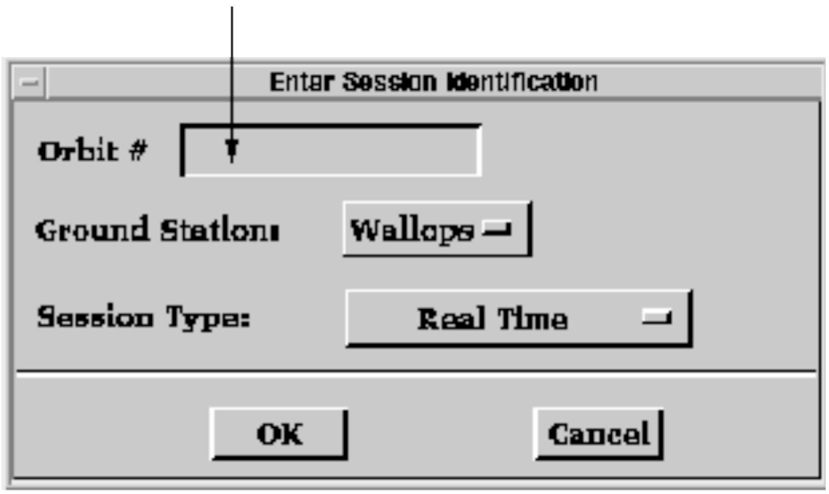
- Create an acronym by taking the first letter of each wc

Monosyllabic Words:

- Take the initial letter and all subsequent consonant
- Make double letters single
- If >4 letters remain, retain the fifth letter if it is part of a functiona cluster (th, ch, sh) - otherwise, truncate from the riç
- Delete the fourth letter if silen

Polysyllabic Words:

- Take the entire first syllable
- If the second syllable starts with a consonant cluster, add
- If the first syllable is a prefix and the second syllable starts with a vowel, add the second syllabl
- Make final double consonants single
- Truncate to four letters (but always retain the entire first syllabl



Inverse shadowing indicates field for data entry.

Orbit #

Ground Station: Wallops

Session Type: Real Time

Figure 4-5. Visual Indication of Data-Entry Field

4.1.6.3.2 Provide a sufficient number of lines and line length to support entry/editing tasks.

4.1.6.3.3 Permit users to change previous entries by delete and insert ("cut and paste") methods. If direct character substitution ("typeover") is included, make the data changes consistently available wherever character substitution is required.

4.1.6.3.4 For cancellation of data entry, have the computer confirm completion of a transaction with a message stating that the cancellation was successful or that there was an error.

4.1.6.3.5 When the user requests changes or deletion of the data, offer the option of maintaining the old value before making the change.

4.1.6.4 Provide the user with the means for selection and entry (e.g., point to and click on; move to and select) of a position on a display, or of a displayed item.

4.1.6.5 To aid the user in entering data in a hierarchic structure with various sections and levels of detail, provide computer aids, such as:

- Question-and-answer dialogues or form filling to maintain data relationships.
- Arrows on flow charts that automatically connect lines.
- Indicators of current positions when panning a map.

4.1.6.5.1 For orientation, consider displaying a defined data structure with branches and levels labeled for reference.

4.1.6.5.2 For complex data structures, provide computer prompts, so that the user can make appropriate data entries at different levels.

4.2 Alphanumerics

The display of continuous text, numbers, and combined alphanumeric codes raises numerous design and layout issues. Designers and developers should have a detailed understanding of user needs for alphanumeric displays before proceeding with any particular approach.

4.2.1 Continuous Text

Reading from a computer screen can be noticeably slower than reading from paper. The challenge in designing and displaying continuous text is to support the user's reading speed and comprehension.

4.2.1.1 If the user needs to read continuous text from the screen, present at least four lines of text at a time, displayed in mixed upper and lower case.

4.2.1.2 Present text in wide lines containing between 40 and 60 characters per line. Avoid narrow columns of short lines.

4.2.1.2.1 Make the display space wide enough to present full lines of text (Figure 4-6). Do not require horizontal scrolling to "uncover" hidden text.

4.2.1.3 Number paragraphs, and separate them by at least one blank line.

4.2.2 Letter Combinations and Special Characters

4.2.2.1 Do not use restricted alphabetic sets for alphabetical data entry.

4.2.2.2 When *special characters* are chosen for keying (e.g., @, /, =, #), select characters that do not require frequent keyboard shifting.

4.2.3 Numbers

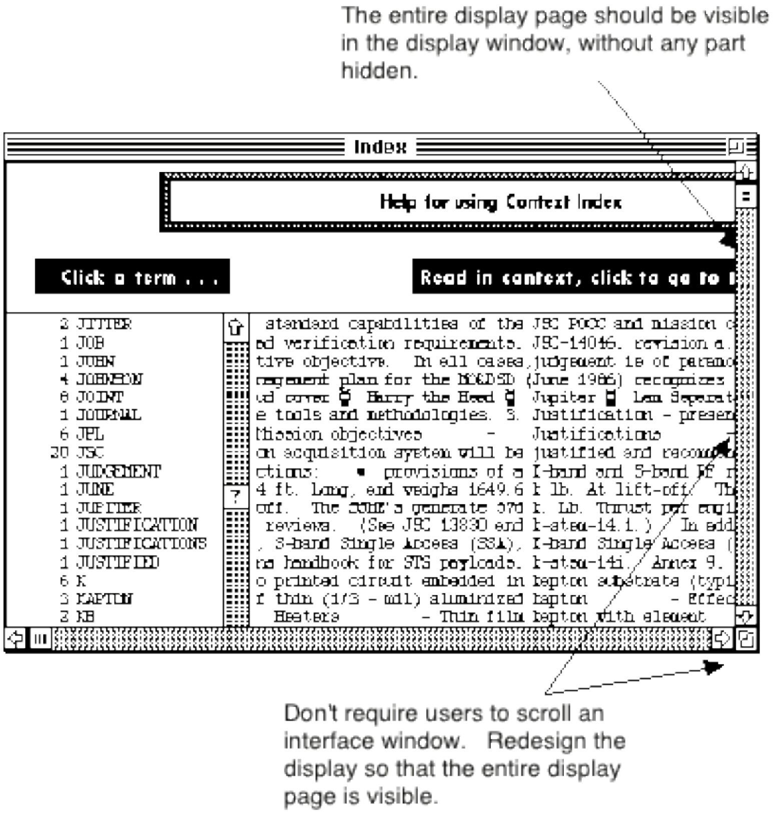
Present data in digital form only if the user needs specific numerical values.

4.2.3.1 Use six or fewer characters in numeric codes.

4.2.3.2 Use Arabic (not Roman) numerals when numbering items in a list.

4.2.3.3 Right justify integers for ease of viewing and scanning.

4.2.3.4 Give the user the option to enter or omit the decimal point at the end of an integer. The system should recognize an entry of "45" and an entry of "45." as equivalent.



The entire display page should be visible in the display window, without any part hidden.

```
 2 JITTER           standard capabilities of the JSC POCC and mission c
 1 JOB             ed verification requirements. JSC-14046. revision a.
 1 JOHN            tive objective.  In all cases, judgement is of paramo
 4 JOHNSON         nagement plan for the MO&DSD (June 1986) recognizes
 8 JOINT           ed cover ☐ Harry the Head ☐ Jupiter ☐ Lem Separat
 1 JOURNAL         e tools and methodologies. 5. Justification - presen
 6 JPL             Mission objectives      -     Justifications     -
20 JSC             on acquisition system will be justified and recom
 1 JUDGEMENT       ctions:    •  provisions of a K-band and S-band RF r
 1 JUNE            4 ft. long, and weighs 1649.6 k lb. At lift-off, Th
 1 JUPITER         off.  The SSME's generate 370 k. Lb. Thrust per engi
 1 JUSTIFICATION    reviews.  (See JSC 13830 and K-atsu-14.1.)  In add
 1 JUSTIFICATIONS  , S-band Single Access (SSA), K-band Single Access (
 1 JUSTIFIED       ns handbook for STS payloads. K-atsu-14i.  Annex 9.
 6 K               o printed circuit embedded in kapton substrate (typi
 3 KAPTON          f thin (1/3 - mil) aluminized kapton         - Effec
 2 KB                Heaters         - Thin film kapton with element
```

Don't require users to scroll an interface window. Redesign the display so that the entire display page is visible.

Figure 4-6. No Horizontal Scrolling in Text Displays

4.2.3.4.1 If a decimal point is required for data processing, program the computer to append one as needed.

4.2.3.4.2 When displayed in columns, decimal numbers should be decimal aligned.

4.2.3.5 Permit the user the option of entering or omitting *leading zeros* for general numeric data. In a field that is four characters long, "45" should be recognized by the system without requiring the user to enter "0045."

4.2.3.6 Separate long numbers into groups of 3 or 4 digits. Use either standard separators or spaces. For example, social security numbers and telephone numbers use hyphens; bank account numbers are often divided by spaces.

4.2.4 Scales

Use a linear scale for numerical data, not the more difficult to interpret logarithmic scale or non-linear scales.

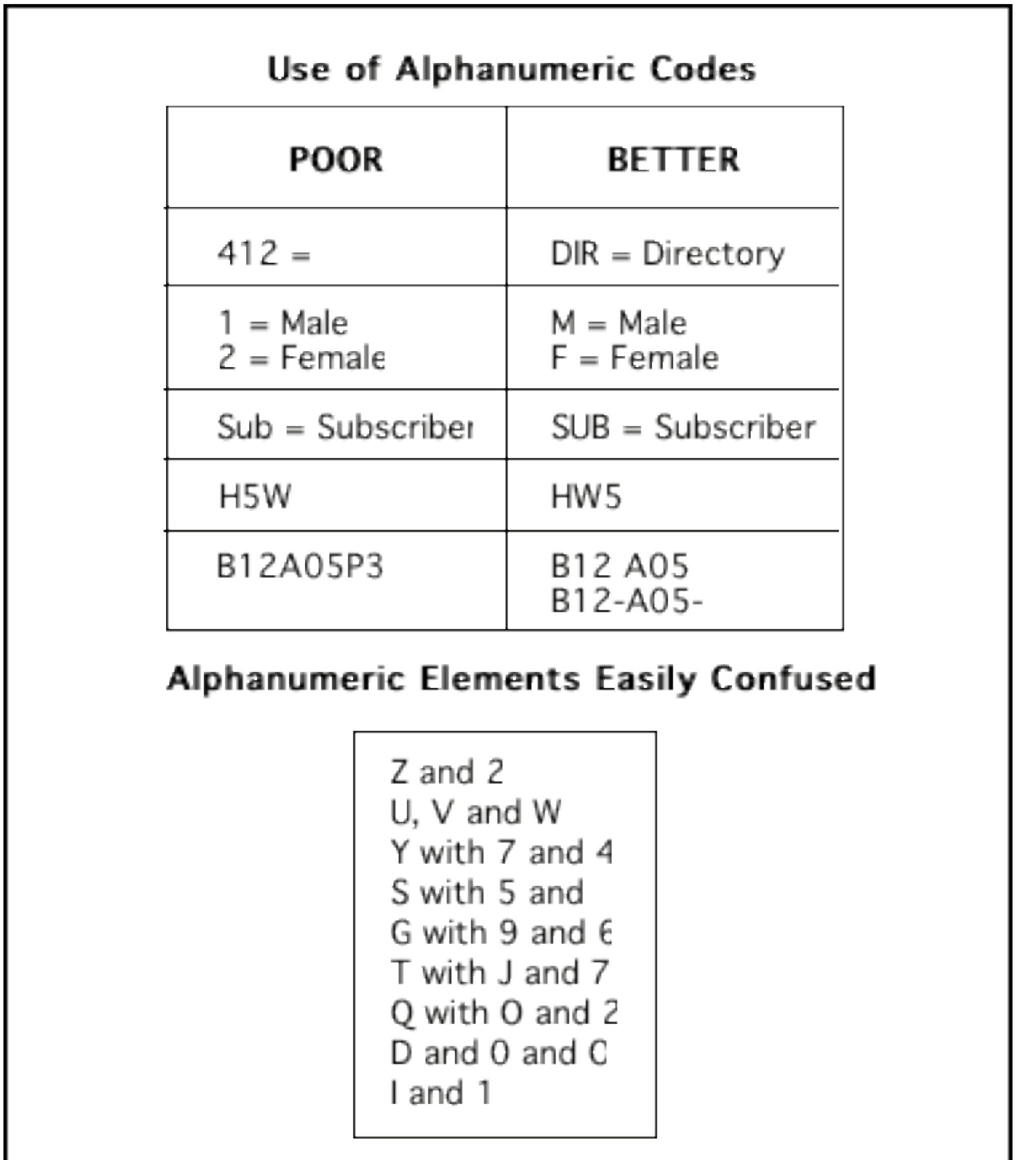
4.2.4.1 Begin scaling numeric data at zero.

4.2.4.2 Use a familiar, meaningful approach to determining scale intervals. For example, use standard intervals of 1, 2, 5, or 10 to label scale divisions.

4.2.5 Alphanumeric Codes

To aid user comprehension of textual and numeric codes, such as acronyms and abbreviations, use meaningful codes in preference to arbitrary codes.

4.2.5.1 Do not choose frequently confused characters and character pairs (Figure 4-7). For example, the letters O and I should not be used in codes because they can be confused with the numbers zero (0) and one (1). Similarly, the number 8 can be confused with the letters B and S.



Use of Alphanumeric Codes

POOR	BETTER
412 =	DIR = Directory

1 = Male
2 = Female
	M = Male
	F = Female

Sub = Subscriber	SUB = Subscriber
H5W	HW5
B12A05P3	B12 A05 B12-A05-

Alphanumeric Elements Easily Confused

Z and 2
U, V and W
Y with 7 and 4
S with 5 and
G with 9 and 6
T with J and 7
Q with O and 2
D and 0 and O
I and 1

Figure 4-7. Alphanumeric Coding

4.2.5.2 When the user must recall alphanumeric codes, limit the code to 5 characters.

4.2.5.3 Group letters and numbers together, rather than interspersing letters and numbers.

4.2.5.3.1 Group characters in blocks of 3 to 5 characters, separated by at least one blank space or other separating character.

4.2.5.4 Use either all upper case or all lower case for alphanumeric codes.

4.2.5.5 Use punctuation in alphanumeric codes only when the code may be confused with a word.

4.2.5.6 When designing alphanumeric commands, consider the effects of possible typographical errors. Ask whether mistyping a command will produce another valid but undesired command.

4.2.5.6.1 Prevent the entry of inappropriate characters into a field (e.g., an alphabetic character into a field reserved for numeric characters).

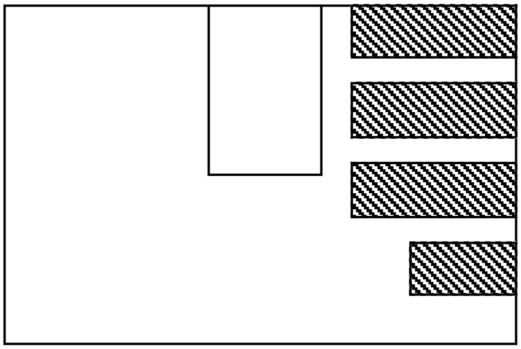
4.3 Graphics

Displaying data in graphical formats can often, but not always, aid the user's visual detection and comprehension of relationships within the data. If the displayed data reflect spatial or temporal relationships, consider displaying them graphically instead of using numerical tables or textual descriptions. Because the benefits of graphics are task dependent, any planned graphical display should be tested for usability.

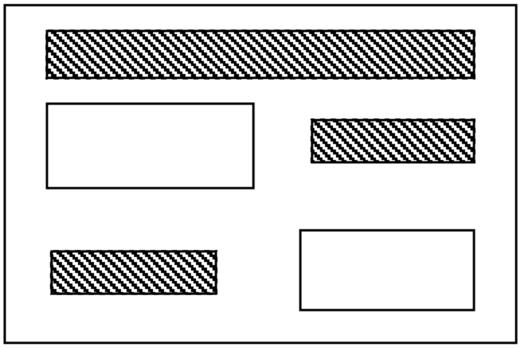
4.3.1 Visual Balance

To make displays readily scannable, balance their elements in as organized a pattern as possible, and maintain that organization across the application. Base the organizing principle on users' expectations and on their understanding of the logic underlying the displayed information.

4.3.1.1 Balance the left side of the display against the right side, and the top against the bottom. Consider using a layout grid as an aid to achieving both visual and conceptual balance of displayed elements.



Unbalanced



Balanced

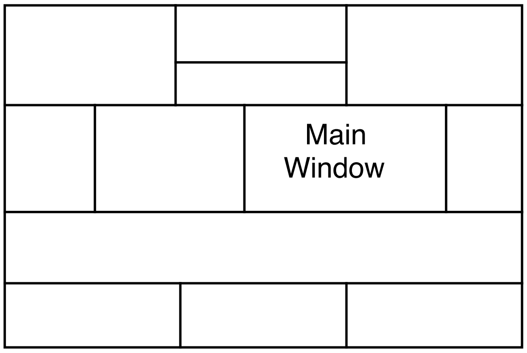
4.3.1.1.1 Use symmetry (an even distribution of elements to the left and right of center) to convey stability.

4.3.1.1.2 If using an asymmetric layout to emphasize contrast or to convey movement, balance the distribution of displayed elements through the use of factors such as color, size, and shape.

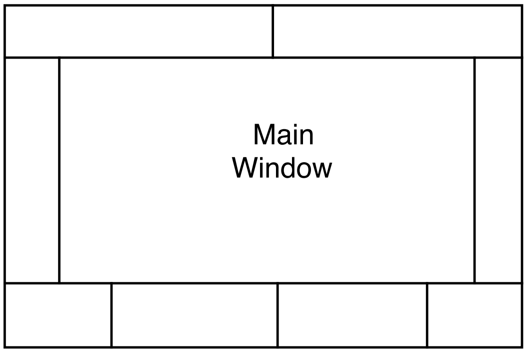
4.3.1.2 Display short messages in wide, shallow spaces.

4.3.1.3 Display graphics in square or slightly rectangular areas. Use recommended proportions of length to width (e.g., 1:1, 1:1.414, 1:1.618). Avoid irregular shapes.

4.3.1.4 Frame the central display area with smaller functional elements:



Poor Layout

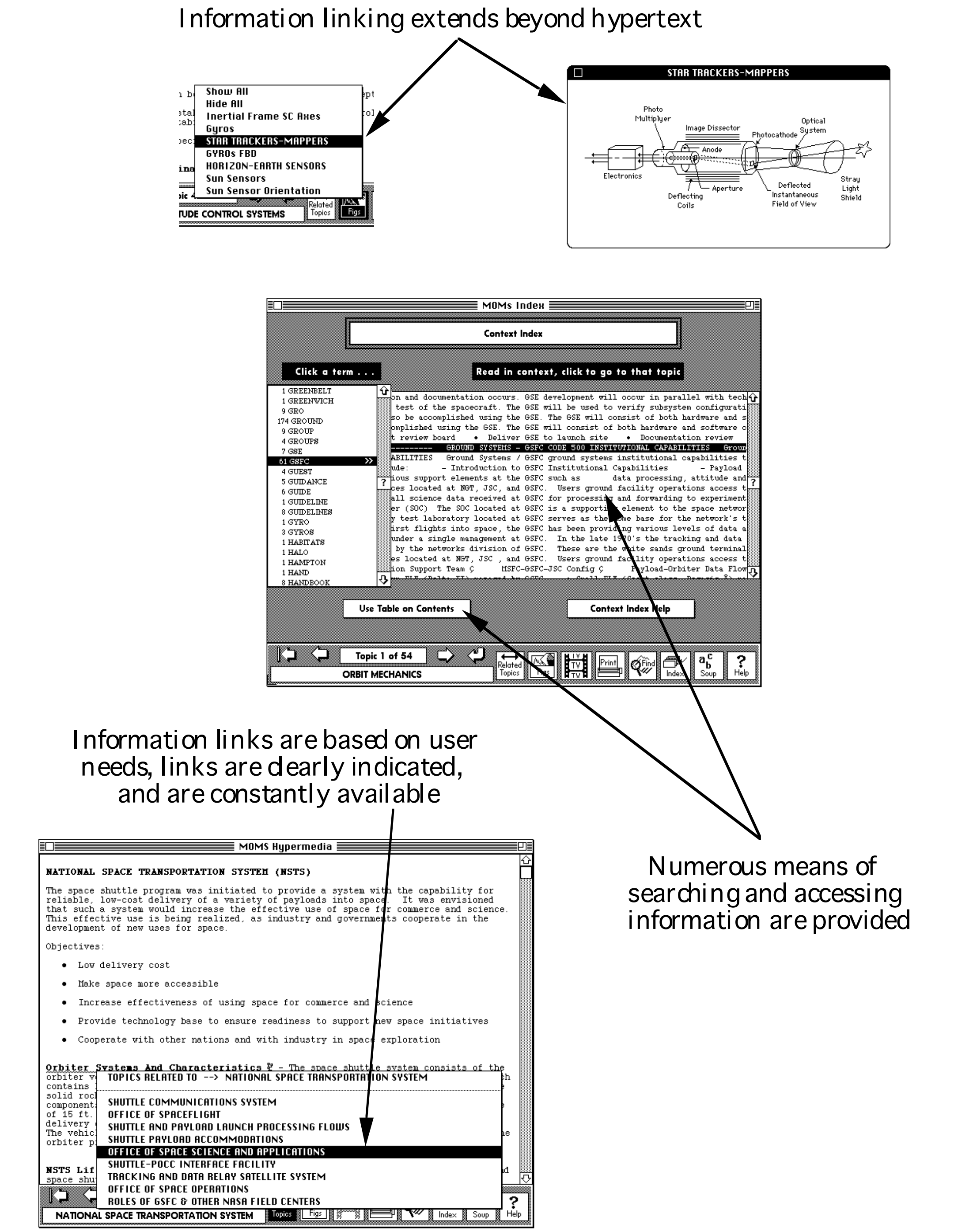


Layout Improved by Framing

4.3.1.5 Leave at least one character space between the contents of a window and the window borders.

4.4 Hypertext and Hypermedia

Hypertext documents are based on a design metaphor that likens the screen to a deck of cards or a series of planes through which the user is able to scroll or pan. The primary features of a hypertext system are nodes (e.g., text, graphics, sound, video) and links between nodes (Figure 4-8). In hypermedia systems, the user controls traversal among nodes.



Information linking extends beyond hypertext

1	GREENBELT
1	GREENWICH
9	GRO
174	GROUND
9	GROUP
4	GROUPS
7	GSE
61	GSFC
4	GUEST
5	GUIDANCE
6	GUIDE
1	GUIDELINE
8	GUIDELINES
1	GYRO
3	GYROS
1	HABITATS
1	HALO
1	HAMPTON
1	HAND
8	HANDBOOK

```
on and documentation occurs. GSE development will occur in parallel with tech
 test of the spacecraft. The GSE will be used to verify subsystem configurati
so be accomplished using the GSE. The GSE will consist of both hardware and s
omplished using the GSE. The GSE will consist of both hardware and software c
t review board    •  Deliver GSE to launch site    •  Documentation review
----------    GROUND SYSTEMS - GSFC CODE 500 INSTITUTIONAL CAPABILITIES   Groun
ABILITIES   Ground Systems / GSFC ground systems institutional capabilities t
ude:       - Introduction to GSFC Institutional Capabilities       - Payload
ious support elements at the GSFC such as       data processing, attitude and
ces located at NGT, JSC, and GSFC.  Users ground facility operations access t
all science data received at GSFC for processing and forwarding to experiment
er (SOC)  The SOC located at GSFC is a supporting element to the space networ
y test laboratory located at GSFC serves as the home base for the network's t
irst flights into space, the GSFC has been providing various levels of data a
under a single management at GSFC.  In the late 1970's the tracking and data
 by the networks division of GSFC.  These are the white sands ground terminal
es located at NGT, JSC , and GSFC.  Users ground facility operations access t
ion Support Team ç      MSFC-GSFC-JSC Config ç      Payload-Orbiter Data Flow
```

Information links are based on user needs, links are clearly indicated, and are constantly available

Numerous means of searching and accessing information are provided

NATIONAL SPACE TRANSPORTATION SYSTEM (NSTS)

The space shuttle program was initiated to provide a system with the capability for reliable, low-cost delivery of a variety of payloads into space. It was envisioned that such a system would increase the effective use of space for commerce and science. This effective use is being realized, as industry and governments cooperate in the development of new uses for space.

Objectives:

- Low delivery cost
- Make space more accessible
- Increase effectiveness of using space for commerce and science
- Provide technology base to ensure readiness to support new space initiatives
- Cooperate with other nations and with industry in space exploration

Orbiter Systems And Characteristics - The space shuttle system consists of the orbiter v contains solid roc component of 15 ft. delivery The vehic orbiter p

NSTS Lif space shu

TOPICS RELATED TO --> NATIONAL SPACE TRANSPORTATION SYSTEM
SHUTTLE COMMUNICATIONS SYSTEM
OFFICE OF SPACEFLIGHT
SHUTTLE AND PAYLOAD LAUNCH PROCESSING FLOWS
SHUTTLE PAYLOAD ACCOMMODATIONS
OFFICE OF SPACE SCIENCE AND APPLICATIONS
SHUTTLE-POCC INTERFACE FACILITY
TRACKING AND DATA RELAY SATELLITE SYSTEM
OFFICE OF SPACE OPERATIONS
ROLES OF GSFC & OTHER NASA FIELD CENTERS
NATIONAL SPACE TRANSPORTATION SYSTEM
Topics
Figs
Index
Soup
? Help

Figure 4-8. Information Linking and Search Capabilities in Hypermedia

4.4.1 Access to Information

Provide alternative means for the user to access information in a hypermedia document (e.g., following links, searching, or using contextual cues).

4.4.2 Links

Links between nodes enable the user to move in a non-linear fashion through the available information. A common problem to be avoided is the presence of so many links that the user becomes overwhelmed.

4.4.2.1 Base the creation of links between related information on the user's need for information in some particular context.

4.4.2.2 Indicate the presence of links by use of a link marker (such as an icon) or by highlighting.

4.4.2.3 Present linked graphical materials with appended text defining the graphical material and its text links.

4.4.2.4 To the extent possible, provide links across media (i.e., in addition to text linking, use graphics linking, sound linking, and video linking).

4.4.3 User Orientation

To avoid user disorientation in navigating hypermedia documents, employ (1) maps or browsers that indicate the user's position within the network, or (2) tags, markers, or milestones that represent locations within the network.

4.4.4 Collaborative Authoring

In collaborative authoring of hypermedia documents, use an authoring system that provides ways to read, link, and edit the document while allowing authors to protect their work from unauthorized access or changes.

References

4.0	Carlow (1992); Galitz (1992); Horton (1990); Marcus (1992); Mayhew (1992); Shneiderman (1992); Smith and Mosier (1986)
4.1	Carlow (1992)
4.1.1	Carlow (1992); Galitz (1993); Smith and Mosier (1986)
4.1.1.1	Carlow (1992); Smith and Mosier (1986)
4.1.1.1.1	Carlow (1992); Smith and Mosier (1986)
4.1.1.1.1.1	Carlow (1992)
4.1.1.1.1.2	Carlow (1992)
4.1.1.1.2	Carlow (1992)
4.1.1.2	Carlow (1992)
4.1.1.3	Carlow (1992); Mitchell and Saisi (1987); Thurman (1995); Thurman and Mitchell (1995)
4.1.1.3.1	Carlow (1992); Smith and Mosier (1986)
4.1.2	Carlow (1992); Smith and Mosier (1986)
4.1.3	Dondis (1973); Siebert and Ballard (1992); Tullis (1981, 1988)
4.1.3.1	Bailey (1982); Carlow (1992); Galitz (1993); Shneiderman (1992); Smith and Mosier (1986); Tullis (1981, 1988)
4.1.3.1.1	Carlow (1992); Smith and Mosier (1986)
4.1.3.1.2	Carlow (1992); Smith and Mosier (1986)
4.1.3.1.3	Carlow (1992); Smith and Mosier (1986)
4.1.3.1.4	Carlow (1992)
4.1.3.1.5	Carlow (1992); Smith and Mosier (1986)
4.1.3.1.6	Smith and Mosier (1986)
4.1.3.2	Carlow (1992)
4.1.3.2.1	Carlow (1992)
4.1.3.2.1.1	Carlow (1992)
4.1.4	Galitz (1993); Shneiderman (1992)
4.1.4.1	Carlow (1992)
4.1.4.2	Carlow (1992)
4.1.4.3	Carlow (1992)
4.1.4.4	Bailey (1982); Carlow (1992)
4.1.4.4.1	Carlow (1992)
4.1.5	Galitz (1993); Mayhew (1992); Meister (1985); Shneiderman (1992); Smith and Mosier (1986)
4.1.5.1	Smith and Mosier (1986)
4.1.5.2	Smith and Mosier (1986)
4.1.5.3	Carlow (1992); Smith and Mosier (1986)
4.1.5.3.1	Carlow (1992); Smith and Mosier (1986)
4.1.5.3.2	Smith and Mosier (1986)
4.1.5.3.3	Carlow (1992); Smith and Mosier (1986)
4.1.5.3.4	Carlow (1992); Smith and Mosier (1986)
4.1.5.3.5	Carlow (1992); Smith and Mosier (1986)
4.1.5.3.6	Carlow (1992)
4.1.5.4	Galitz (1993); Smith and Mosier (1986)
4.1.6	Galitz (1993)
4.1.6.1	Galitz (1993); Smith and Mosier (1986)

4.1.6.2	Smith and Mosier (1986)
4.1.6.2.1	Smith and Mosier (1986)
4.1.6.2.2	Smith and Mosier (1986)
4.1.6.3	Smith and Mosier (1986)
4.1.6.3.1	Smith and Mosier (1986)
4.1.6.3.2	Smith and Mosier (1986)
4.1.6.3.3	Smith and Mosier (1986)
4.1.6.3.4	Smith and Mosier (1986)
4.1.6.3.5	Smith and Mosier (1986)
4.1.6.4	Smith and Mosier (1986)
4.1.6.5	Galitz (1993)
4.1.6.5.1	Galitz (1993)
4.1.6.5.2	Galitz (1993)
4.2	Smith and Mosier (1986); Marcus (1992); Shneiderman (1992)
4.2.1	Gould and Grischkowsky (1984); for a discussion, see Shneiderman (1992, pp. 443-444).
4.2.1.1	Carlow (1992); Smith and Mosier (1986)
4.2.1.2	Fowler and Stanwick (1995)
4.2.1.2.1	Carlow (1992)
4.2.1.3	Carlow (1992)
4.2.2	Shneiderman (1992)
4.2.2.1	Smith and Mosier (1986)
4.2.2.2	Smith and Mosier (1986)
4.2.2.3	Smith and Mosier (1986)
4.2.2.4	Smith and Mosier (1986)
4.2.3	Smith and Mosier (1986)
4.2.3.1	Smith and Mosier (1986)
4.2.3.2	Smith and Mosier (1986)
4.2.3.3	Mayhew (1992)
4.2.3.4	Smith and Mosier (1986)
4.2.3.4.1	Smith and Mosier (1986)
4.2.3.4.2	Mayhew (1992)
4.2.3.5	Smith and Mosier (1986)
4.2.3.6	Mayhew (1992)
4.2.4	Smith and Mosier (1986)
4.2.4.1	Smith and Mosier (1986)
4.2.4.2	Smith and Mosier (1986)
4.2.5	Carlow (1992)
4.2.5.1	Carlow (1992)
4.2.5.2	Carlow (1992)
4.2.5.3	Carlow (1992)
4.2.5.3.1	Carlow (1992)
4.2.5.4	Carlow (1992)
4.2.5.5	Carlow (1992)
4.2.5.6	Smith and Mosier (1986)
4.2.5.6.1	Carlow (1992)

4.3	Shneiderman (1992); Smith and Mosier (1986)
4.3.1	Horton (1990)
4.3.1.1	Dumas (1988); Horton (1990)
4.3.1.1.1	Siebert and Ballard (1992)
4.3.1.1.2	Siebert and Ballard (1992)
4.3.1.2	Horton (1990)
4.3.1.3	Horton (1990)
4.3.1.4	Horton (1990)
4.3.1.5	Marcus (1992)
4.4	Marcus (1992); Shneiderman (1992)
4.4.1	Carlow (1992)
4.4.2	Shneiderman (1992)
4.4.2.1	Carlow (1992)
4.4.2.2	Carlow (1992)
4.4.2.3	Carlow (1992)
4.4.2.4	Carlow (1992)
4.4.3	Carlow (1992)
4.4.4	Carlow (1992)

5.0 Guidelines for Interaction Styles and Data Protection

Depending on the user's task environment, one or more styles of interaction may be appropriate. The varieties of interaction styles include alphanumeric dialogs (e.g., fill-in forms) as well as graphical dialogs (e.g., menu-selection and direct-manipulation). Many user interfaces combine textual and graphical styles. The key design challenges are to match the interaction style to the user's task domain and to protect the integrity of the data.

5.1 Fill-In Forms

Fill-in forms are used to enter predefined items into labeled fields. These screens resemble paper forms which are completed by filling in the blanks. The screens provide necessary cues so that novice users can correctly determine what must be keyed, enter the appropriate information, and then review it. Major issues include design of entry fields and transactions.

5.1.1 Compatible Forms

Design forms so that data items are ordered in a way that is familiar to the user.

5.1.1.1 If transcriptions from source documents are used in data entry, ensure that form-filling displays are compatible with such documents in terms of item ordering, data grouping, labeling, and so forth.

5.1.2 Entry-Field Basics

Design data-entry fields consistently within and across applications. That is, use consistent approaches to labeling, prompting, highlighting, justification, and spacing.

5.1.2.1 Label each data field to inform users of entries to be made.

5.1.2.1.1 Keep labels close to associated data fields; separate them by at least one space.

5.1.2.1.2 Ensure consistency in field labels by using the same label for the same kind of data entry.

5.1.2.1.3 Protect field labels from keyed entry by having the cursor skip over them when the user is spacing or tabbing.

5.1.2.1.4 Use descriptive wording, standard terms, codes and/or abbreviations when labeling data fields. For more clarity, employ additional cues in a data field label.

Example: Birth Date mm/dd/yy: _ _/_ _/_ _

5.1.2.1.5 Include a unit of measurement as part of the field label when that unit (e.g., $, °, mph) is part of a particular data field. The units of measurement employed should be familiar to the user. When alternative units of measurement are used, provide space to distinguish the units that are entered.

Example: Weight: _ _ _ (lb/kg) _ _

5.1.2.2 In form-filling, require one explicit entry action at the end of the transaction sequence for the entry of logically related items, to avoid the separate entry of each item.

5.1.2.2.1 Allow the user to enter multiple data items without keying special separator or delimiter characters.

5.1.2.2.2 Permit the user to review, cancel, or backup multiple data items entered as a single transaction.

5.1.2.3 If a delimiter is to be used, be consistent in the employment of a standard character. Choose a character that does not require shifting keys or one that does not occur as part of any data entry.

5.1.2.3.1 Do not require users to remove delimiters or otherwise enter keystrokes for all positions within a variable length field.

5.1.2.4 Select a standard means of prompting users for keyed entry, such as highlighting the entry field with beveled edges or shadows.

5.1.2.4.1 Ensure that the method used for highlighting is visually different from screen error messages.

5.1.2.4.2 Highlight entry fields consistently across screens.

5.1.2.5 Clearly distinguish between required and optional entry fields. For example, dashes might indicate required fields, and dots, optional fields.

Example

Requirement No.: _ _ _ _ _ _
User: _ _ _ _ _ _
Date: ....
Time: ....

5.1.2.5.1 Provide a means for indicating when a fixed or maximum length is specified for data entry (Figure 5-1).

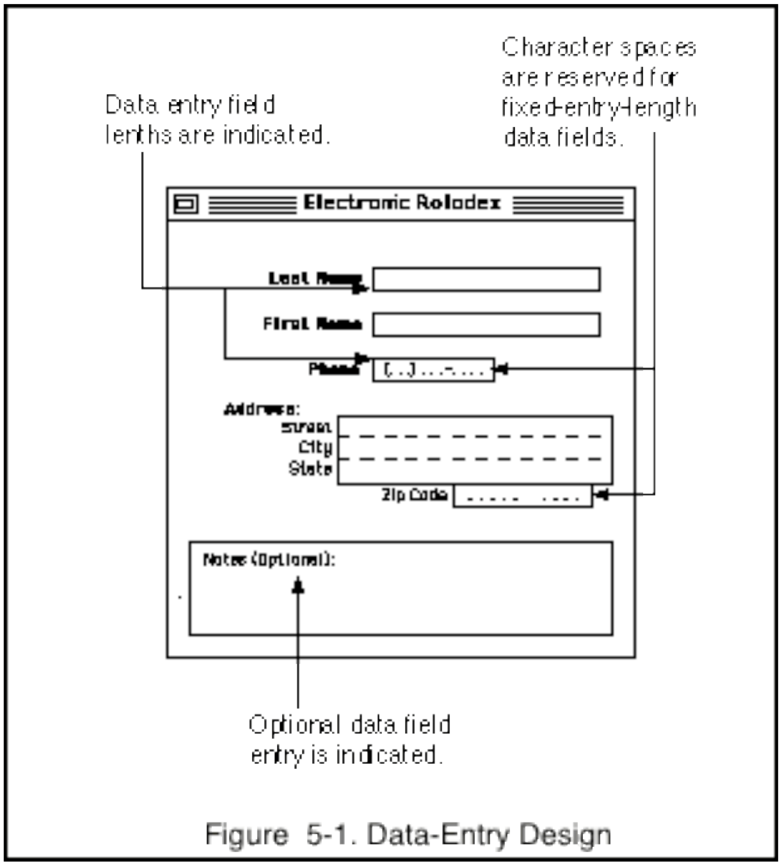


Figure 5-1. Data-Entry Design

5.1.2.5.2 Use field delineation cues, such as coding the labels of required and optional entry fields.

5.1.2.5.3 If the data entry does not completely fill the markers because data entry length is variable, ignore the remaining field markers in computer processing.

5.1.2.5.4 Do not accomplish data entry by overwriting a set of characters within a field.

5.1.2.6 To speed data entry, offer default values that can be defined for data entry in a specific task.

5.1.2.6.1 In the event of a series of defined default values, permit the user to accept defaults for all entries or to default until reaching the next required, non-default entry.

5.1.2.6.2 When it is not possible to predict what default values will be useful, permit the user to define, change, or remove default values for any data-entry field, without changing default definitions for subsequent transactions.

5.1.2.6.3 At the beginning of a data-entry transaction, display default values in the appropriate fields.

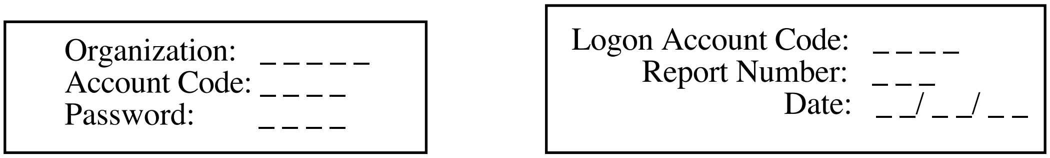
5.1.2.7 Provide automatic updating to free the user from entering the same data twice.

5.1.2.8 Limit data items to 5-7 characters when the user must code data or enter numbers.

5.1.2.8.1 If the data items must be longer, partition them into shorter clusters for entry and display. For example, ten digit telephone numbers are partitioned into three groups: XXX-XXX-XXXX.

5.1.2.9 Use a consistent approach to the justification of data entry fields.

5.1.2.9.1 Left justify the caption and the entry fields, with one space between the longest caption and the entry field. Alternatively, right-justify the captions and left-justify the entry fields.

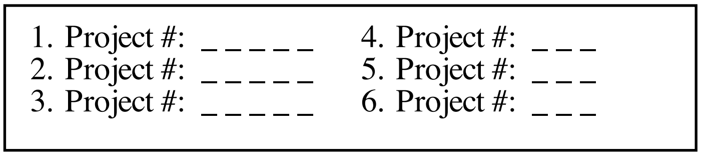


```
Organization:  _ _ _ _ _
Account Code:  _ _ _ _
Password:      _ _ _ _
```

```
Logon Account Code:  _ _ _ _
    Report Number:  _ _ _
             Date:  _ _/_ _/_ _
```

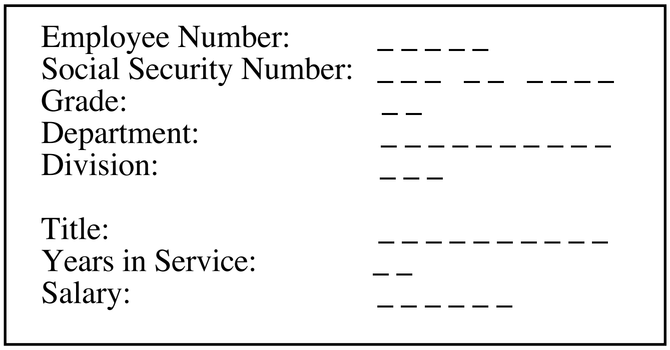
Alternative Justification of Caption and Entry Fields

5.1.2.10 For horizontal viewing, provide a minimum of five spaces between the longest entry field in one column and the left-most caption in the adjacent column. For vertical viewing, provide one blank line between groups of related information.



```
1. Project #: _ _ _ _ _     4. Project #: _ _ _
2. Project #: _ _ _ _ _     5. Project #: _ _ _
3. Project #: _ _ _ _ _     6. Project #: _ _ _
```

Horizontal Spacing of Entry Fields



```
Employee Number:          _ _ _ _ _
Social Security Number:   _ _ _  _ _  _ _ _ _
Grade:                    _ _
Department:               _ _ _ _ _ _ _ _ _ _
Division:                 _ _ _

Title:                    _ _ _ _ _ _ _ _ _ _
Years in Service:         _ _
Salary:                   _ _ _ _ _ _
```

Vertical Spacing of Entry Fields

5.1.3 Command Keystrokes

Across interaction styles, when speed of command input is important, use command keystrokes (i.e., a limited number of keystrokes combined with pressing a command key to access a command language term).

5.1.4 Function Keys

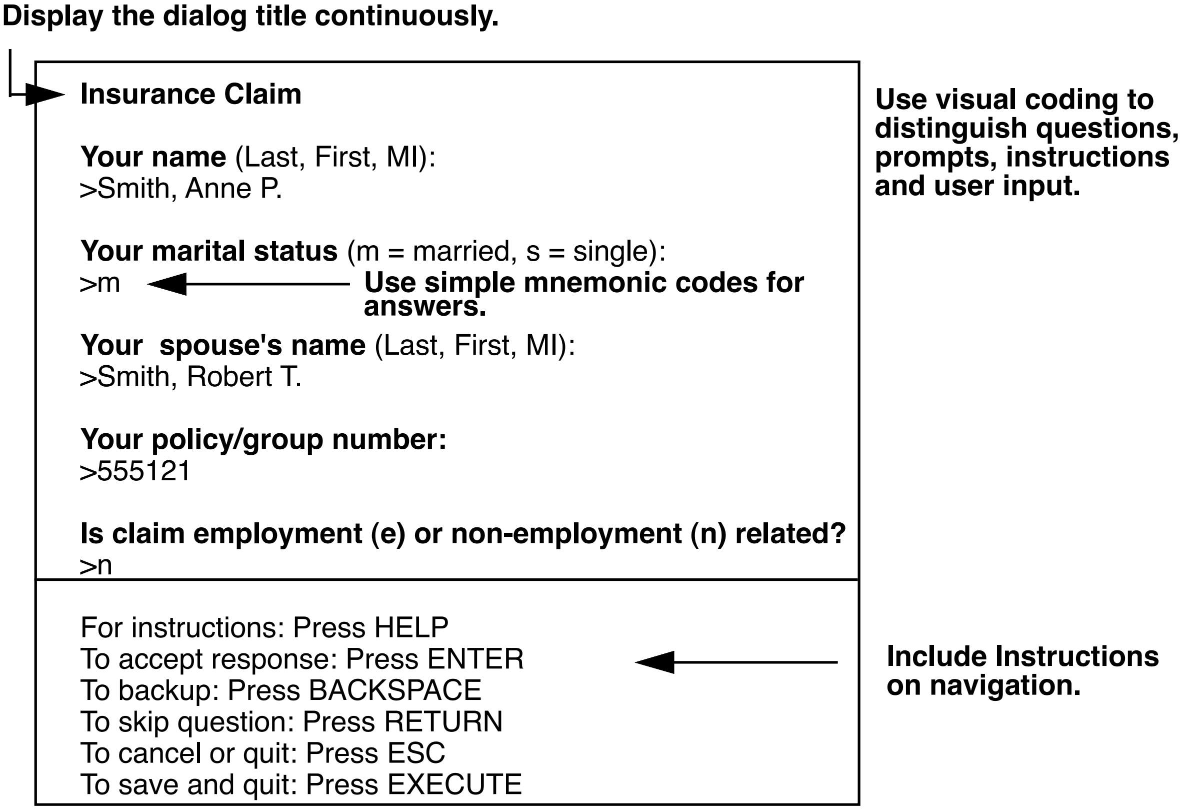
Use function keys for tasks with unique control entries or as an adjunct to other interaction styles for functions that occur frequently, that must be made quickly, and that must be made with minimal syntax errors.

5.2 Question-and-Answer

A question-and-answer interface is a combination of menu and fill-in form interfaces in a dialog style (Figure 5-2). The dialog proceeds in a step-by-step continual interaction. Typically, prompting is included with the question. Sometimes questions and answers are scrolled on the screen, enabling the user to view much of the dialog. In other cases, new questions individually refresh the screen, omitting from view previous questions and answers.

Question-and-answer dialogs are used for the following situations:

- Tasks require routine data entry.
- Data items are known and ordering can be constrained.
- Users are novices and new to the system.
- Computer response time is relatively fast.



Display the dialog title continuously.

Insurance Claim

Your name (Last, First, MI):
>Smith, Anne P.

Your marital status (m = married, s = single):
>m

Use simple mnemonic codes for answers.

Your spouse's name (Last, First, MI):
>Smith, Robert T.

Your policy/group number:
>555121

Is claim employment (e) or non-employment (n) related?
>n

For instructions: Press HELP
To accept response: Press ENTER
To backup: Press BACKSPACE
To skip question: Press RETURN
To cancel or quit: Press ESC
To save and quit: Press EXECUTE

Use visual coding to distinguish questions, prompts, instructions and user input.

Include Instructions on navigation.

Figure 5-2. Question-and-Answer Interface Design

(Source: Adapted from Mayhew, 1992)

5.2.1 Wording of Questions

Ensure questions are in clear, simple language. The grammatical form should be consistent throughout.

5.2.1.1 Avoid negatives, such as "If the claim is not related to an accident, check here." Phrase the question in a positive manner, such as "Check here if the claim is related to an accident."

5.2.2 Mnemonic Codes and Abbreviations for Answers

Use simple mnemonic codes for answers to questions. The user should not have to perform several translations to correctly answer a question. For example, when the user is asked for marital status (Figure 5-2), the user indicates married or single through a simple *mnemonic code*.

5.2.2.1 When a question requires only two possible answers as in the question, "Are you an employee (e) or a visitor (v)," provide mnemonic codes such as "e" for "employee" and "v" for "visitor."

5.2.2.2 Minimize typing requirements by requesting abbreviated rather than spelled out answers, such as "m" for married and "y" for yes. Avoid capital letters when not necessary (for example, do not require "M" for married and "Y" for yes).

5.2.3 Recapitulation of Prior Answers

Display interrelated, computer-posed questions so that answers to previous questions are visible to the user. This often helps the user in answering questions that follow.

5.2.4 Visible Titles

Display the dialog title continuously, despite questions scrolling off the screen as the dialog progresses (Figure 5-2).

5.2.5 Visual Coding of Dialog Parts

Use visual cues and white space so that the user can distinguish between questions, prompts, instructions and user input. Capital versus lower-case letters, indentations, and spatial location are some other methods that can be used to identify dialog parts. In Figure□5-2, bold text and prompts of the same size differ from the arrow prompt and smaller size letters for the answers.

5.2.6 Navigation Instructions

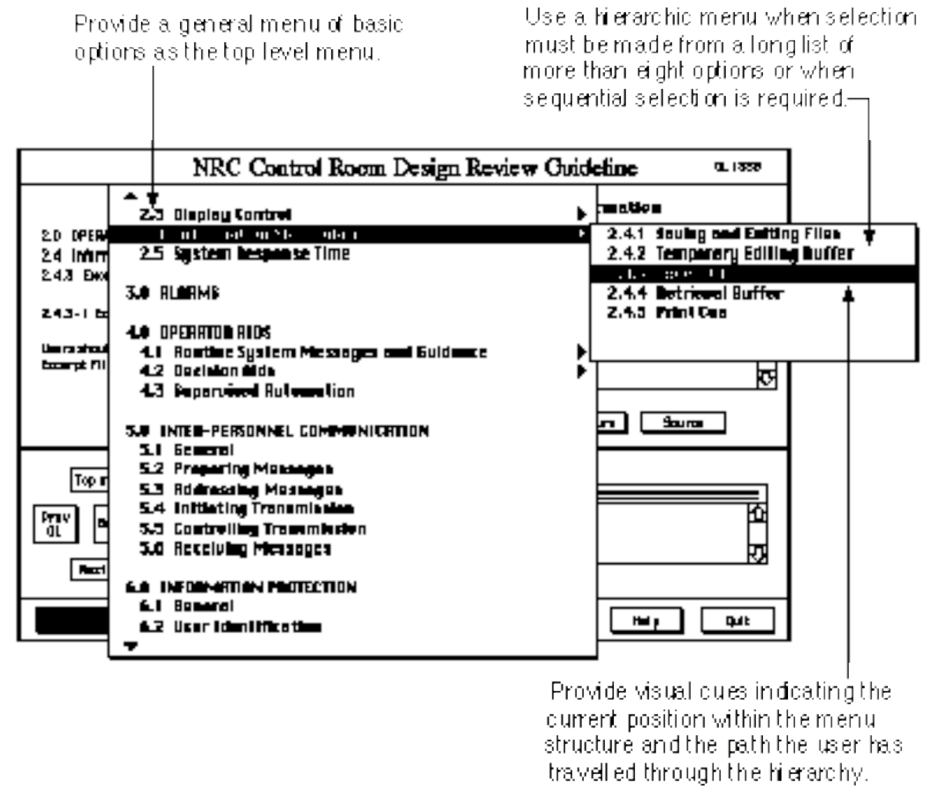
Include instructions for using navigation capabilities; otherwise the user may never discover and make use of them.

5.2.7 Scanning Capabilities

Provide the user with the capability to return to and edit previous answers, scan through future questions, quit the session before completion of dialog, and re-enter the dialog to the point where the user left off.

5.3 Menus

The key objectives of menu design are to minimize the user's search-and-selection time and to optimize the user's ability to navigate through the menu structure (Figure 5-3).


NRC Control Room Design Review Guideline
2.3 Display Control
2.5 System Response Time
3.0 ALARMS
4.0 OPERATOR AIDS
4.1 Routine System Messages and Guidance
4.2 Decision Aids
4.3 Supervised Automation
5.0 INTER-PERSONNEL COMMUNICATION
5.1 General
5.2 Preparing Messages
5.3 Addressing Messages
5.4 Initiating Transmission
5.5 Controlling Transmission
5.6 Receiving Messages
6.0 INFORMATION PROTECTION
6.1 General
6.2 User Identification
2.4.1 Saving and Editing Files
2.4.2 Temporary Editing Buffer
2.4.4 Retrieval Buffer
2.4.5 Print Que
Source
Help
Quit

Provide a general menu of basic options as the top level menu.

Use a hierarchic menu when selection must be made from a long list of more than eight options or when sequential selection is required.

Provide visual cues indicating the current position within the menu structure and the path the user has travelled through the hierarchy.

Figure 5-3. Menu Structure and Visual Cues

5.3.1 General

Consider using menus when

- User tasks involve selecting from a constrained set of alternative actions.
- User tasks require only occasional data entry.
- Users have little experience or training.
- Computer response is reasonably fast.
- Commands from a large command set are used infrequently.

5.3.1.1 Locate menus and menu options consistently when they appear in different displays.

5.3.1.2 Provide a means for the experienced user to bypass the menu structure by using direct keyboard commands. Display alternative keyboard commands (*accelerators*) to the right of the menu option label. Do not use arbitrary numbers for accelerator codes.

5.3.1.3 If logical grouping of options is not possible, display up to eight options that are appropriate to any particular transaction.

5.3.1.3.1 If selection must be made from a list of more than eight options or if sequential selection is required, consider using a hierarchic menu structure (Figure 5-3).

5.3.1.3.2 If logical categories can be established for a set of more than eight options, as many as 64 categorized options may be presented in one menu.

5.3.1.4 Dim (or gray out) unavailable or invalid options.

5.3.2 Phrasing Menu Options

Use direct, unambiguous wording that reflects the actions to be executed. Use verbs or verb phrases (not nouns or noun phrases) to denote actions (e.g., edit, view, insert).

5.3.2.1 Where possible, use wording that is familiar to users, but do not use familiar terms to mean something different from their common usage in the user community.

5.3.3 Formatting Menu Options

Menu options can be formatted linearly in a vertical or horizontal list or spatially in a circle, rectangle, or other geometric shape (e.g., *pie menus*). The following guidelines apply to the layout, organization, and grouping of vertically formatted menu options.

5.3.3.1 Format the menu list as a single column, wide enough to accommodate the widest option plus its keyboard accelerator.

5.3.3.2 Arrange menu and submenu options in separate, cascading columns, with the options listed under each other, and left-justified.

5.3.3.3 Use logical sequence, criticality, or frequency of use to establish the ordering of options in a brief menu.

5.3.3.3.1 Make the most likely selection in a menu list the default option.

5.3.3.3.2 Within hierarchic menus, permit immediate access to critical or frequently selected options.

5.3.3.4 If basing menu organization on frequency of use, place the following options at the bottom of the menu:

- Less frequently used options.
- Destructive commands, such as Delete or Exit.

5.3.3.5 Within a longer menu, group logically related options, and draw a solid line between groups. Make the solid line the same color as the option labels.

5.3.3.6 If there is no inherent logical order in a set of menu options, alphabetize the options, but do not place options for opposing actions adjacent to each other. For example, the Delete option should not be next to the Save option.

5.3.4 Pull-down Menus

Pull-down menus appear in response to the user's selection of a menu option from the menubar. They are also called "drop-down" menus. Pull-down menus should be used when space is limited, when users need to see the menu options only when selecting them, and when required information on the screen will not be obscured by the menu. Pull-down menus include various types of options (e.g., command actions, settings or toggles, submenus or cascades).

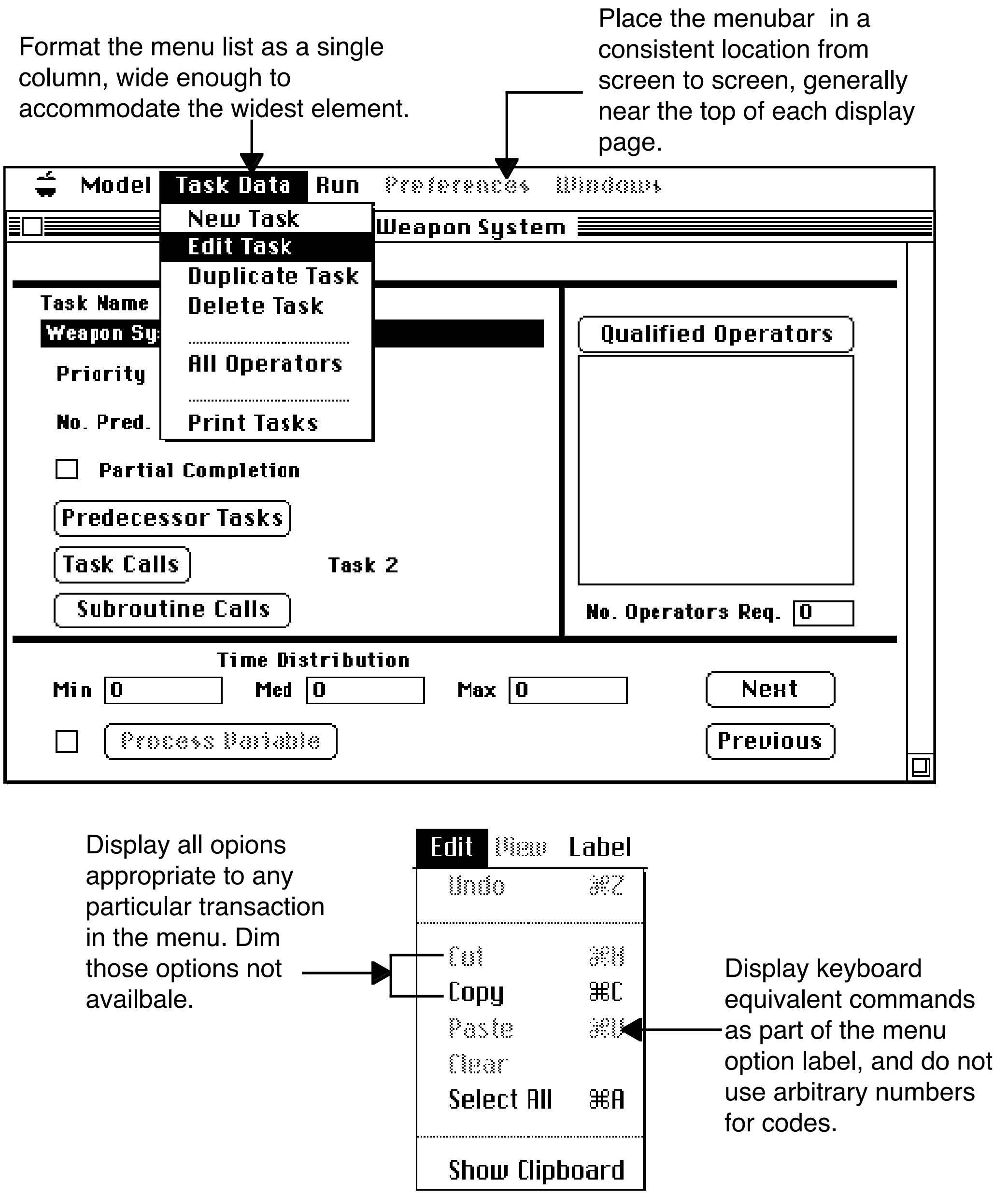
5.3.4.1 Use pull-down menus to open top-level menus listed in the menubar (Figure□5-4).

5.3.4.2 Place the pull-down directly below the option selected from the menubar. The higher-level option serves as the title for the pull-down.

5.3.4.3 Present the pull-down in the same foreground and background colors used in the menubar. (These colors should contrast sufficiently with the background of the application area/screen body.)

5.3.4.4 Outline the pull-down with a border or drop shadow.

5.3.4.5 To the extent possible, present all options to minimize scrolling.



Format the menu list as a single column, wide enough to accommodate the widest element.

Place the menubar in a consistent location from screen to screen, generally near the top of each display page.

Display all opions appropriate to any particular transaction in the menu. Dim those options not availbale.

Display keyboard equivalent commands as part of the menu option label, and do not use arbitrary numbers for codes.

Figure 5-4. Pull-Down Menu Format

5.3.4.6 Present the options in initial capital letters followed by lower-case letters ("mixed case") or in self-explanatory graphics (e.g., fill-in patterns, colors), as appropriate to the user's task.

5.3.4.7 Display a pull-down menu until the user makes a selection.

5.3.4.8 Do not place instructions in pull-down menus.

5.3.4.9 If a pull-down option leads to a second-level, cascading pull-down, follow the option label with a right-pointing arrow. Position the cascading pull-down to the right of the highlighted option in the previous pull-down.

The recommended number of levels of cascading pull-downs varies between one (Apple) and three (DEC windows). To support the user's sense of orientation within the menu structure, it is best to minimize the number of levels.

5.3.5 Pop-up Menus

Each window or displayed object may have an associated pop-up menu that appears, on demand, in response to a user action (e.g., point and click on a designated screen area or menu icon). Visually, a pop-up menu resembles a pull-down menu, but it is not associated with the top-level menus listed in the menubar. A pop-up menu typically contains five to ten options presented in a vertical listing. In some environments, pop-up menus may cascade and may include keyboard accelerators.

5.3.5.1 Provide some indication or cue to the existence of a pop-up menu (for example, highlight the portion of the display that can be selected to access the hidden menu; provide a textual message indicating that a hidden menu is available; or change the shape of the cursor when it is located in a "clickable" or selectable area).

5.3.5.2 Place a pop-up menu directly below the pointer used to select it and near the object or higher-level menu that is being manipulated.

5.3.5.3 Display a title for each pop-up menu.

5.3.5.4 When an option in a pop-up menu leads to a cascading menu, place a right-pointing triangle after the option label.

5.3.5.5 Make the pop-up distinct from the screen background by giving it a contrasting, yet complementary, background or by giving it a solid-line border.

5.3.5.6 Highlight an option selected from a pop-up menu. In particular, continue to highlight an option that leads to a cascading menu. The highlighted option serves as the title for the cascading menu.

5.3.5.7 Because of the variation in usage and implementation of pop-up menus across different environments, check your UI software style guide before incorporating pop-up menus in your application's user interface.

5.3.6 Option-Button Menus

When screen space is limited, use an option-button menu to present a brief set of specialized menu options (Figure 5-5).

5.3.6.1 Implement an option-button menu so that it opens like a pull-down menu, allowing the user to see all of the possible options.

5.3.6.2 Distinguish an option-button menu from a pushbutton.

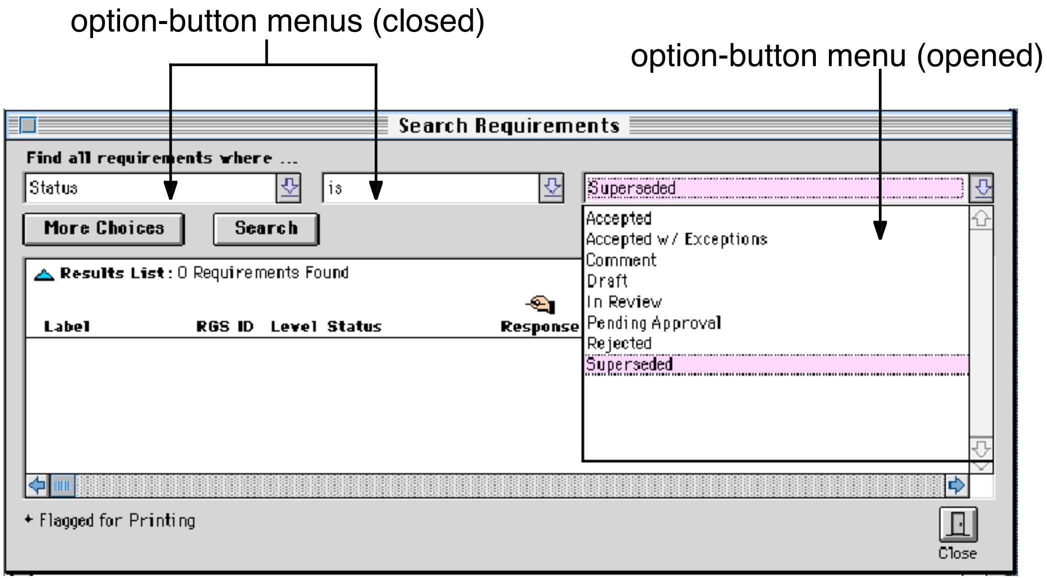


Figure 5-5. Option-Button Menus

5.3.7 Graphic Menus

Graphic menus are sometimes called palettes. A *palette* is a set of unlabeled symbols, typically presented within small rectangles . Symbols may be icons, patterns, characters, or drawings that represent an operation. When activated, a symbol provides access to a software tool (i.e., a set of specialized software functions). Palettes are used widely in drawing and painting packages but have proliferated even in word-processing applications. A major problem for the user is not knowing what the symbols stand for.

5.3.7.1 Ensure that symbols are self-explanatory since they are not labeled.

5.3.7.2 Because selection of a symbol (tool) sends the user into a *mode*, display a reminder of the mode that is in effect (e.g., by changing the shape of the pointing cursor).

5.3.7.3 Allow the user to move and re-size the palette.

5.3.7.4 Reflect settings for the active window in the palette.

5.3.8 Tear-off Menus

A *tear-off menu* can be "torn" from the menubar and moved to another location on the screen where it can remain on display. Tear-off menus are also called "tacked" or "pushpin" menus.

5.3.8.1 Keep a tear-off menu posted so that the user can make multiple selections before dismissing it.

5.3.8.2 Use a graphic tear-off menu, instead of a fixed palette, to save display space and provide greater flexibility.

5.3.9 Aiding Menu Navigation

One of the biggest problems for users of menuing systems is getting lost in the menu structure. A key objective of menu design is to support the user's ability to navigate both forward and backward through the menu structure. The following aids to menu *navigation* can be used singly or in combination, depending on the nature of the user's goals, tasks, and information requirements.

5.3.9.1 Provide graphical or textual aids to assist users in maintaining their orientation within the underlying menu structure. (Graphic aids include designated font styles, line types, or colors for different menu levels. Textual aids include numbering schemes and descriptive titles).

5.3.9.1.1 Provide visual cues indicating the user's current position within the menu structure and the path the user has traveled through a hierarchy of menus (Figure 5-3).

5.3.9.1.2 Consider displaying a small schematic of the menu structure, highlighting the path taken as the user proceeds. The schematic serves the function of a road map.

5.3.9.1.3 Consider providing a textual list of the options already selected, as an alternative to the high-lighted menu map.

5.3.9.2 Display the primary navigational aid continuously or make it available from a pull-down menu.

5.3.9.3 Keep the top-level menu options available in the menubar.

5.3.9.4 In a hierarchical menu, allow the user to return to the next higher menu level or to the top-level menu with a single control action.

5.3.9.5 Provide visual indicators, such as ellipses (...), to make menu options that branch to other submenus distinguishable from menu options that will immediately perform an operation.

5.3.9.6 Provide support for various user search strategies.

5.3.9.6.1 Provide the capability to move both forward and backward through the menu structure.

5.3.9.7 Allow the experienced user to use optional shortcuts, such as *direct access, type ahead, jump ahead* or *user-developed macros*, to speed up traversal of the menu structure.

5.3.9.8 Consider various strategies for repositioning search automatically, when the user has reached a terminal node in a hierarchical menu without finding the target menu option.

5.3.9.8.1 Reposition search to the next higher level that affords the widest breadth of selection.

5.3.9.8.2 Provide a capability for users to mark points in the menu structure where they might want to return and a capability to return to such a point by issuing a command.

5.3.9.8.3 Reposition to intermediate "*landmarks*" instead of forcing the user to return to the root node to restart a local search.

5.3.9.9 Sequence menus according to the logical flow of control required by the user's task.

5.3.9.10 Consider allowing users to create their own pathways through the menu structure (e.g., by adding horizontal and diagonal links within the original structure).

5.4 Direct Manipulation

For the user, graphical representations are often much easier to retain and manipulate than are textual or numeric representations. By means of *direct manipulation,* the user senses that the displayed environment is acted upon with actions that are rapid, and with results that are continuously visible.

5.4.1 Manipulation Techniques

Techniques of direct manipulation include open/close, minimize/maximize, drag-and-drop, resize, and scroll. Many direct-manipulation tasks are initiated by the user's action of pointing and clicking on screen-control graphics or icons.

5.4.1.1 Continuously display graphical reminders of actions that can be performed.

5.4.1.2 Provide feedback both about the action to be performed and about the action as it is performed.

5.4.1.2.1 If a selection action is to be performed, highlight the selected object.

5.4.1.2.2 If the action is to be a movement, alter the pointing cursor's shape and highlight the object to be moved. Continue to highlight the object as it is moved.

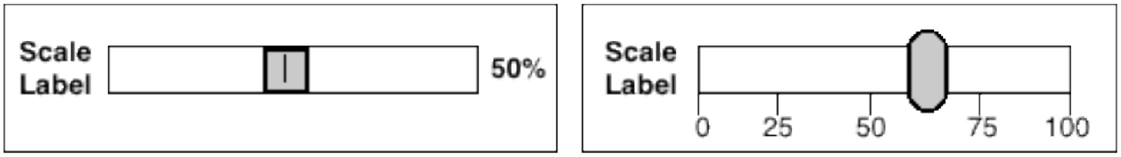
5.4.1.2.3 In a drag-and-drop action, move the entire object as the user drags it; do not allow the object to lag behind the pointing cursor.

5.4.1.3 Permit the user to cancel a selection before executing the associated action.

5.4.1.4 Provide smooth, even tracking between the screen pointer and the input device (e.g., mouse, trackball).

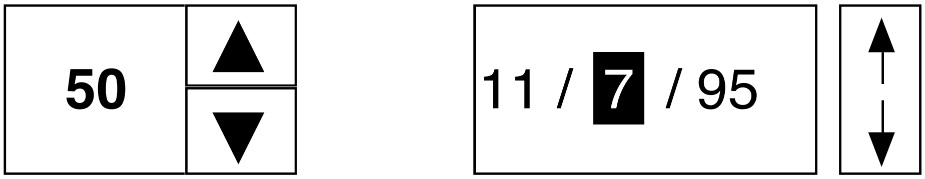
5.4.1.5 Immediately display the results of users' direct manipulations.

5.4.1.6 Consider using a slider bar and button to allow the user to set values represented along a scale.



Examples of Slider Bars

5.4.1.7 Consider using a spin button to allow the user to select from a long list of options that increase or decrease uniformly, in constant intervals.



Examples of Spin Buttons

5.4.2 Browsing

The coordination of multiple windows that can appear, change contents, and close as a result of user activity is a design that is useful for browsing. Techniques for browsing are dependent on the task for which the design is developed. Types of multiple window coordinations and procedures for browsing are as follows:

- Synchronized Scrolling

  The scroll bar of one window is linked to the scroll bar of an adjoining window. When one scroll bar is moved, it causes the other to scroll its window contents. This allows the browser to compare the contents of both windows simultaneously.

- Hierarchical Browsing

  Selecting an item in the first window with a pointing device can lead to the display of an adjoining window that contains more detail. The development of coordinated windows that display different levels of information is regarded as a finer-grained approach.

- Direct Selection

  Pointing at an icon, a word in the text, or a variable name in a program results in the pop up of an adjoining window with information about the icon, word, or variable name.

- Two-Dimensional Browsing

  A map, graphic, photographs, or other image is presented in a high-level view. The details of that view are then simultaneously presented in magnified form in another, larger window.

- Dependent-Windows Opening

  The opening of dependent windows that are conveniently located on a display is one option of browsing. When browsing a program, opening a main procedure results in a dependent set of procedures opening up as well.

- Dependent-Windows Closing

  With a single action, the closing of a window results in the closing of all its dependent windows. The advantage of dependent-window closing is the one-step process of removing those windows from the screen that are no longer applicable, such as dialogs, messages, and help windows.

- Save Or Open Window State

  Saving the state of a display as it appears on the screen with all the windows intact is a feature useful for browsing. The implementation of a *Save By…* menu item to the *File* menu permits the user to access that state for future reference.

5.5 Data Protection

Whenever a proposed user action will interrupt a transaction sequence, it is necessary to provide the automatic protection of data. If potential data loss cannot be prevented, warn the user and require confirmation prior to implementation.

Data integrity can also be advanced by limiting access, by building protection around dangerous operations, and by maintaining security procedures. Providing an "undo" function and effective on-line help are major ways to protect data and transaction sequences from the unintended consequences of possible user actions.

5.5.1 Sequence Control

The various interaction styles support the completion of transactions and the sequencing of transactions. A transaction is one complete exchange of information between the user and the computer. Transactions include keystrokes and their immediate display as well as command entries (e.g., Save, Print) and their execution. A sequence is a series of user actions combined with computer logic that results in the initiation, interruption, or termination of transactions.

Key design objectives are to maintain consistency among control actions, limit the need for control actions, reduce loading on the user's memory, and provide flexibility for adaptation to a range of user needs. Sequence control should be in the hands of the user, with system control subordinated to user control.

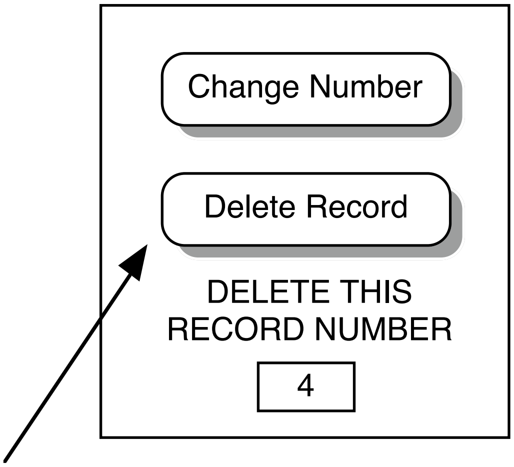
5.5.1.1 Use the structure of the user's task to design a sequence of related transactions. A sequence of transactions should form a logical unit from the user's point of view and should provide the control options needed at any point.

5.5.1.1.1 Provide transaction options that match expected user goals and tasks.

5.5.1.2 Provide multiple, flexible means of sequence control so that users can accomplish necessary transactions involving data entry, display, and processing, or can obtain guidance as needed.

5.5.1.2.1 Give users more than one method of controlling scrolling and paging.

5.5.1.3 Permit users to control transaction sequencing by explicit action. Defer computer processing until that explicit user action is taken.

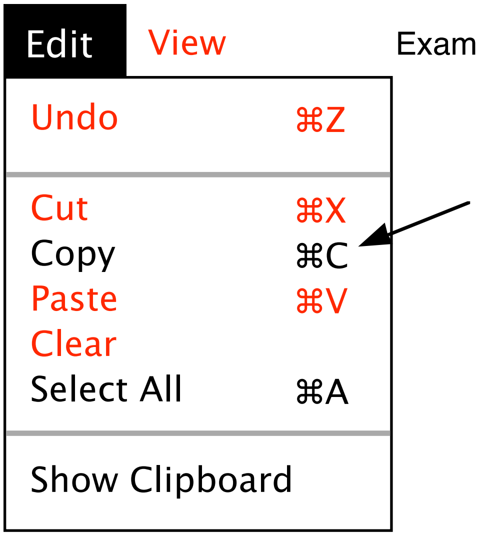


Example:
Explicit user action is required, especially for data destructive operations

5.5.1.3.1 Provide a means to override control lockout due to processing.

5.5.1.4 Provide immediate system acknowledgement of every control entry.

5.5.1.5 Permit users to key a sequence of commands or option codes as a single massed or "stacked" command entry.



Example: Massing commands should be provided allowing users to string commands together, without waiting for system processing to be completed prior to issuing command strings. For example, holding down the command key (⌘) and typing "AC" should 1) select all, and 2) copy selection.

5.5.1.6 Provide interrupt or abort functions to terminate transactions.

5.5.1.6.1 Consider the need for the following types of interrupts, in addition to backup and undo:

- Cancel -- abort a command or operation.
- Review -- return to the first display in a sequence and permit the user to step through the sequence.
- Restart -- cancel all entries made in the transaction sequence and return to the beginning of the sequence.
- End -- conclude a repetitive transaction sequence.
- Pause -- interrupt the transaction sequence.
- Continue -- resume a paused transaction sequence without any change to data entries or control logic.
- Suspend -- preserve current transaction status when the user leaves the system, and permit resumption of work at that point on re-entry.

5.5.1.6.2 If different types of transaction interrupts are provided, designate each interrupt function as a separate control action with a distinct name.

5.5.1.6.3 Display paused or suspended status.

5.5.1.7 Provide prompts for resumption of a transaction sequence.

5.5.1.7.1 Permit users to search for specific line numbers and literal strings of alphanumeric characters.

5.5.1.8 Minimize the number of screens required to complete a transaction.

5.5.2 Access Security

Include a simple, prompted log-on process to protect data from unauthorized access (Figure□5-6).

5.5.2.1 Provide for user selection of passwords, and use software to enforce periodic changing of passwords.

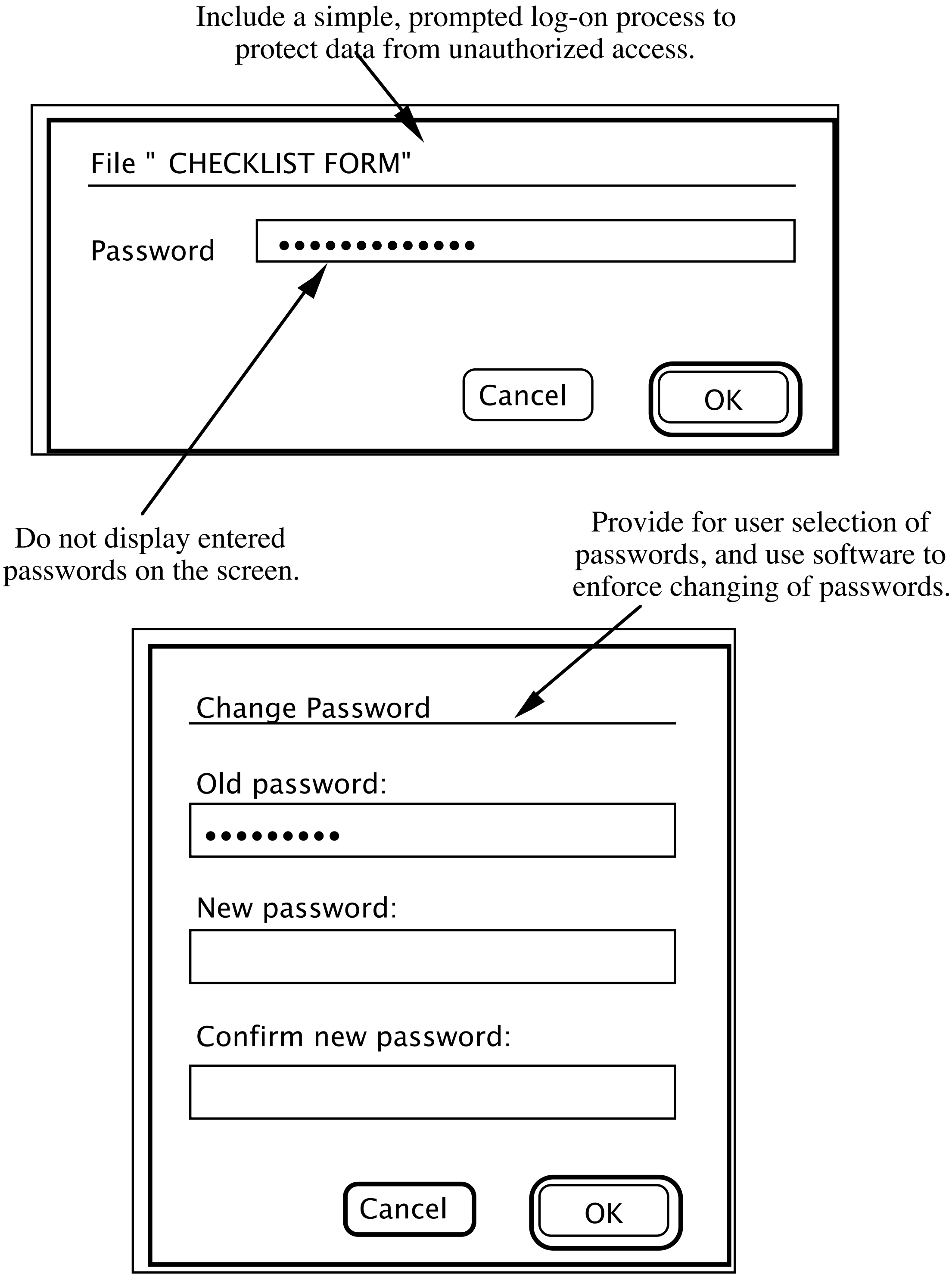
5.5.2.1.1 Do not display entered passwords on the screen.

5.5.2.2 Establish user authorization for data display/entry/change at initial log-on. Do not require additional authorization when the user attempts to display data, enter data, or make changes to data.

5.5.2.3 Limit the number and rate of unsuccessful attempts to log-on.

5.5.2.4 To protect highly sensitive information, require another authentication of the user's identity after ten minutes of inactivity.

5.5.2.5 When a record of data access is required, have the system maintain the records automatically (i.e., do not expect the user to keep records manually or through the system).



Include a simple, prompted log-on process to protect data from unauthorized access.

Do not display entered passwords on the screen.

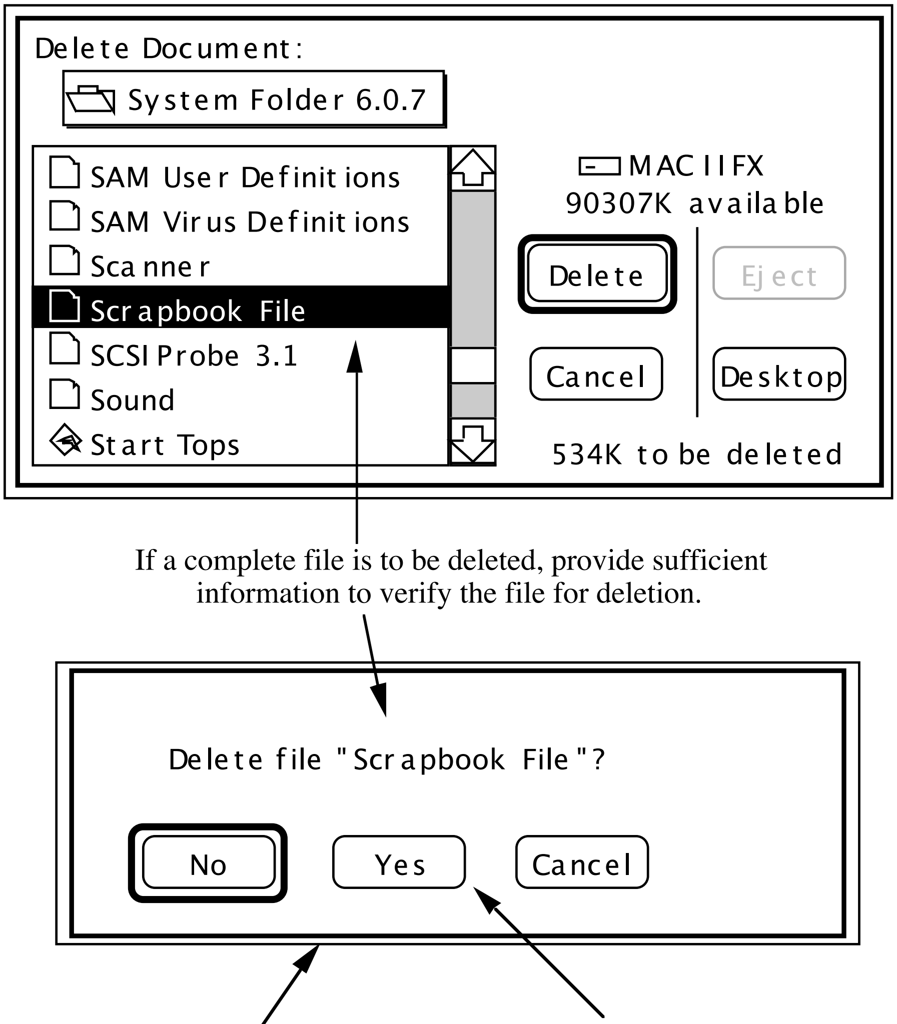
Provide for user selection of passwords, and use software to enforce changing of passwords.

Figure 5-6. Data Protection: Access Security

5.5.3 Dangerous Operations

Maintain the integrity of the data. Build protection around dangerous operations, such as accidental file deletion (Figure 5-7).

5.5.3.1 Prior to command execution, require users to confirm that they want to perform a critical, potentially hazardous, or potentially destructive command (including commands that would destroy stored data).


Delete Document:

If a complete file is to be deleted, provide sufficient information to verify the file for deletion.

Delete file "Scrapbook File"?

Require users to confirm that they want to perform a critical, potentially hazardous, or potentially destructive command before execution.

Do not make a dangerous condition/option a default. Require the user to select it.

Figure 5-7. Data Protection: Dangerous Operations

5.5.3.1.1 If a complete file is to be deleted, provide sufficient information for the user to confirm that the file is to be deleted (e.g., file name, description, size, creation and modification dates).

5.5.3.1.2 Label a confirm action clearly and distinctively.

5.5.3.1.3 Make explicit the effects that will follow a confirmation.

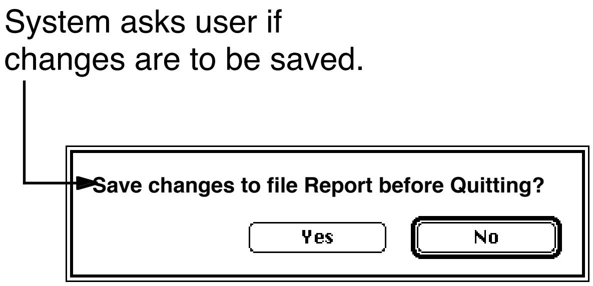
5.5.3.1.4 Require the user to wait for computer prompting before entering a confirmation (i.e., do not buffer the confirm affirmation).

5.5.3.3 Require the user to select a dangerous operation instead of making it a default. (Note: This may be a deviation from design consistency, but it is necessary in the interest of data integrity.)

5.5.3.4 Deactivate potentially destructive function keys or other devices when they are not needed.

5.5.3.5 Protect display-formatting features, such as field labels and delimiters, from accidental change by users.

5.5.3.6 Provide clear and consistent procedures for potentially dangerous transactions (e.g., global operations or those involving data change or deletion (Figure-5-8)).



System asks user if changes are to be saved.

Save changes to file Report before Quitting?

Figure 5-8. Data-Entry-Change Confirmation

5.5.4 Information Security

When displayed data are classified or controlled, display a prominent indication of their status.

5.5.4.1 When displayed data must be protected, maintain computer control over the display (e.g., do not allow users to change a "read-only" field), and display the status of the screen (e.g., read-only).

5.5.4.2 When classified or controlled data are to be displayed at an uncontrolled workstation, provide a rapid means of temporarily suppressing the screen display if privacy is threatened.

References

5.0	Mayhew (1992); Shneiderman (1992)
5.1	Mayhew (1992)
5.1.1	Mayhew (1992)
5.1.1.1	Mayhew (1992)
5.1.2	Smith and Mosier (1986)
5.1.2.1	Smith and Mosier (1986)
5.1.2.1.1	Smith and Mosier (1986)
5.1.2.1.2	Smith and Mosier (1986)
5.1.2.1.3	Smith and Mosier (1986)
5.1.2.1.4	Smith and Mosier (1986)
5.1.2.1.5	Smith and Mosier (1986)
5.1.2.2	Galitz (1993)
5.1.2.2.1	Galitz (1993)
5.1.2.2.2	Smith and Mosier (1986)
5.1.2.3	Smith and Mosier (1986)
5.1.2.3.1	Carlow (1992)
5.1.2.4	Smith and Mosier (1986)
5.1.2.4.1	Smith and Mosier (1986)
5.1.2.4.2	Smith and Mosier (1986)
5.1.2.5	Carlow (1992); Smith and Mosier (1986)
5.1.2.5.1	Carlow (1992)
5.1.2.5.2	Carlow (1992)
5.1.2.5.3	Smith and Mosier (1986)
5.1.2.5.4	Carlow (1992)
5.1.2.6	Carlow (1992); Smith and Mosier (1986)
5.1.2.6.1	Smith and Mosier (1986)
5.1.2.6.2	Carlow (1992); Smith and Mosier (1986)
5.1.2.6.3	Carlow (1992); Smith and Mosier (1986)
5.1.2.7	Carlow (1992); Smith and Mosier (1986)
5.1.2.8	Carlow (1992); Smith and Mosier (1986)
5.1.2.8.1	Carlow (1992); Smith and Mosier (1986)
5.1.2.9	Galitz (1993)
5.1.2.9.1	Galitz (1993)
5.1.2.10	Galitz (1993)
5.1.3	Carlow (1992)
5.1.4	Carlow (1992)
5.2	Mayhew (1992); Smith and Mosier (1986)
5.2.1	Mayhew (1992); Smith and Mosier (1986)
5.2.1.1	Mayhew (1992); Smith and Mosier (1986)

5.2.2	Mayhew (1992); Smith and Mosier (1986)
5.2.2.1	Mayhew (1992)
5.2.2.2	Mayhew (1992)
5.2.3	Galitz (1993); Smith and Mosier (1986)
5.2.4	Mayhew (1992)
5.2.5	Galitz (1993); Mayhew (1992)
5.2.6	Mayhew (1992)
5.2.7	Mayhew (1992)
5.3	Norman (1991); Shneiderman (1992)
5.3.1	Mayhew (1992); Norman (1991)
5.3.1.1	Shneiderman (1992)
5.3.1.2	Carlow (1992)
5.3.1.3	Carlow (1992); Galitz (1993)
5.3.1.3.1	Carlow (1992)
5.3.1.3.2	Galitz (1993); Paap and Roske-Hofstrand (1986, 1988)
5.3.1.4	Carlow (1992); Shneiderman (1992); for a discussion, see Mayhew (1992), pp. 149-150.
5.3.2	Carlow (1992)
5.3.2.1	Galitz (1992); Shneiderman (1992)
5.3.3	Norman (1991); Mayhew (1992); Shneiderman (1992)
5.3.3.1	Carlow (1992)
5.3.3.2	Galitz (1993)
5.3.3.3	Carlow (1992)
5.3.3.3.1	Carlow (1992)
5.3.3.3.2	Carlow (1992)
5.3.3.4	Fowler and Stanwick (1995); Mayhew (1992); Smith and Mosier (1986)
5.3.3.5	Galitz (1993)
5.3.3.6	Carlow (1992); Mayhew (1992)
5.3.4	Fowler and Stanwick (1995)
5.3.4.1	Carlow (1992)
5.3.4.2	Galitz (1993)
5.3.4.3	Galitz (1993)
5.3.4.4	Galitz (1993)
5.3.4.5	Galitz (1993)
5.3.4.6	Galitz (1993)
5.3.4.7	Apple Computer (1987); Hix and Hartson (1993)
5.3.4.8	Galitz (1993)
5.3.4.9	Galitz (1993)
5.3.5	Fowler and Stanwick (1995); Galitz (1993)
5.3.5.1	Galitz (1993)
5.3.5.2	Galitz (1993)
5.3.5.3	Galitz (1993)
5.3.5.4	Galitz (1993)
5.3.5.5	Galitz (1993)
5.3.5.6	Galitz (1993)
5.3.5.7	Fowler and Stanwick (1995)

5.3.6	Fowler and Stanwick (1995); e.g., OSF (1991)
5.3.6.1	Kobara (1991)
5.3.6.2	Kobara (1991); OSF (1991)
5.3.7	Apple Computer (1987); Hix and Hartson (1993); Horton (1990)
5.3.7.1	Horton (1990)
5.3.7.2	Apple Computer (1987); Hix and Hartson (1993)
5.3.7.3	Apple Computer (1987)
5.3.7.4	Apple Computer (1987)
5.3.8	Fowler and Stanwick (1995)
5.3.8.1	OSF (1991)
5.3.8.2	Apple Computer (1987)
5.3.9	Galitz (1993); Mayhew (1992); Norman (1991); Shneiderman 992)
5.3.9.1	Galitz (1993); Horton (1991); ISO (1993, Part 14); Mayhew (1992); Norman (1991)
5.3.9.1.1	Carlow (1992)
5.3.9.1.2	Galitz (1993); Horton (1991); Mayhew (1992)
5.3.9.1.3	Galitz (1993); Paap and Roske-Hofstrand (1988)
5.3.9.3	Galitz (1993); e.g., Apple Computer (1987); OSF (1991)
5.3.9.4	Carlow (1992); Smith and Mosier (1986)
5.3.9.5	Carlow (1992); Galitz (1993); Smith and Mosier (1986)
5.3.9.6	Carlow (1992); Shneiderman (1992); Smith and Mosier (1986)
5.3.9.6.1	Carlow (1992)
5.3.9.7	Norman (1991); Shneiderman (1992)
5.3.9.8	Norman (1991)
5.3.9.8.1	Norman (1991)
5.3.9.8.2	Norman (1991)
5.3.9.8.3	Norman (1991)
5.3.9.9	Norman (1991)
5.3.9.10	Norman (1991)
5.4	Mayhew (1992); Shneiderman (1992)
5.4.1	Shneiderman (1992); Galitz (1993)
5.4.1.1	Galitz (1993)
5.4.1.2	Galitz (1993)
5.4.1.2.1	Galitz (1993)
5.4.1.2.2	Galitz (1993)
5.4.1.2.3	Galitz (1993)
5.4.1.3	Galitz (1993)
5.4.1.4	Galitz (1993)
5.4.1.5	Shneiderman (1992); Galitz (1993)
5.4.1.6	Fowler and Stanwick (1995)
5.4.1.7	Fowler and Stanwick (1995)
5.4.2	Shneiderman (1992)
5.5	Carlow (1992); Smith and Mosier (1986)
5.5.1	Carlow (1992); Smith and Mosier (1986)
5.5.1.1	Carlow (1992); Smith and Mosier (1986)
5.5.1.1.1	Carlow (1992)

5.5.1.2	Carlow (1992); Smith and Mosier (1986)
5.5.1.2.1	Carlow (1992)
5.5.1.3	Carlow (1992); Smith and Mosier (1986)
5.5.1.3.1	Carlow (1992); Smith and Mosier (1986)
5.5.1.4	Carlow (1992); Smith and Mosier (1986)
5.5.1.5	Carlow (1992); Smith and Mosier (1986)
5.5.1.6	Carlow (1992)
5.5.1.6.1	Carlow (1992)
5.5.1.6.2	Carlow (1992)
5.5.1.6.3	Carlow (1992); Smith and Mosier (1986)
5.5.1.6.1	Carlow (1992); Smith and Mosier (1986)
5.5.1.7	Carlow (1992)
5.5.1.7.1	Smith and Mosier (1986)
5.5.1.8	Smith and Mosier (1986)
5.5.2	Carlow (1992); Smith and Mosier (1986)
5.5.2.1	Carlow (1992); Smith and Mosier (1986)
5.5.2.1.1	Carlow (1992); Smith and Mosier (1986)
5.5.2.2	Carlow (1992); Smith and Mosier (1986)
5.5.2.3	Carlow (1992); Smith and Mosier (1986)
5.5.2.4	Carlow (1992); Smith and Mosier (1986)
5.5.2.5	Carlow (1992); Smith and Mosier (1986)
5.5.3	Carlow (1992)
5.5.3.1	Carlow (1992)
5.5.3.1.1	Carlow (1992)
5.5.3.1.2	Carlow (1992)
5.5.3.1.3	Carlow (1992)
5.5.3.1.4	Carlow (1992)
5.5.3.3	Carlow (1992)
5.5.3.4	Carlow (1992)
5.5.3.5	Carlow (1992); Smith and Mosier (1986)
5.5.3.6	Carlow (1992)
5.5.4	Carlow (1992); Smith and Mosier (1986)
5.5.4.1	Carlow (1992); Smith and Mosier (1986)
5.5.4.2	Carlow (1992); Smith and Mosier (1986)

6.0 Guidelines for Display Control and Window□Design

Resolving issues of data selection and presentation is a key concern of the development team. These display-control issues become even more complex in a windowed environment. Key human-factors issues center on the design of window appearance and behavior (i.e., "look and feel"). A design objective is to minimize time spent in window management or "housekeeping."

Check the style guide that applies to your environment's UI development software to determine how much latitude you have in changing the look and feel of windows in your application.

6.1 Display Control

Designers are expected to determine as many user requirements for specific tasks as possible. Because *all* user requirements cannot be anticipated by the designer, give users the flexibility to tailor their displays on line by controlling data selection, data coverage within a display frame, data updating, and data suppression.

6.1.1 User Control

Support user control by permitting users to control the amount, format, movement, and complexity of displayed data. Within data-protection constraints, users should be allowed to change displayed data or enter new data.

6.1.1.1 Indicate display-control options clearly and appropriately for selection by the user (Figure 6-1). (Determine appropriateness from the task analysis and from discussions with the user group.)

6.1.1.2 To the extent possible (without crowding), include data relevant to the user's current transaction in one display frame or page.

6.1.1.3 Provide an easy and consistent means of moving through the data, such as windowing, panning, paging, or scrolling.

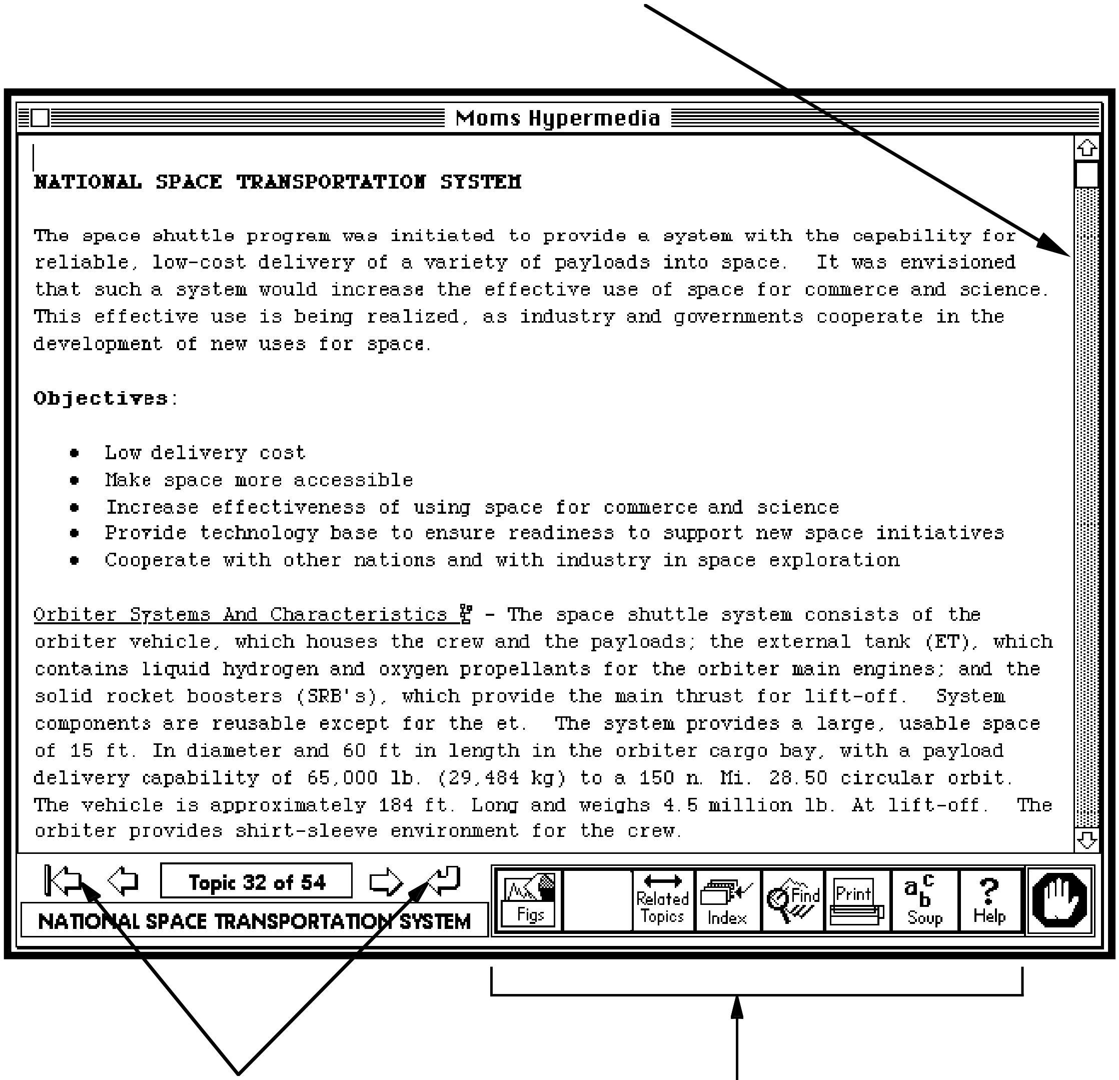
6.1.1.3.1 Use scrolling to support the expert user's search through continuous text. Do not use paging or windowing for this purpose.

6.1.1.3.2 Use paging for the user's search of logically grouped information, such as data forms. Do not use panning or scrolling for this purpose.

6.1.1.3.3 Use up/down arrows to indicate vertical scrolling direction and previous/next or left/right arrows to indicate paging direction.

6.1.1.4 Provide a zoom in/out capability to support the user's detailed examination of graphical displays (e.g., maps, pictures, diagrams).

Scroll bars are used to move through continuous text.



NATIONAL SPACE TRANSPORTATION SYSTEM

The space shuttle program was initiated to provide a system with the capability for reliable, low-cost delivery of a variety of payloads into space. It was envisioned that such a system would increase the effective use of space for commerce and science. This effective use is being realized, as industry and governments cooperate in the development of new uses for space.

Objectives:

- Low delivery cost
- Make space more accessible
- Increase effectiveness of using space for commerce and science
- Provide technology base to ensure readiness to support new space initiatives
- Cooperate with other nations and with industry in space exploration

Orbiter Systems And Characteristics - The space shuttle system consists of the orbiter vehicle, which houses the crew and the payloads; the external tank (ET), which contains liquid hydrogen and oxygen propellants for the orbiter main engines; and the solid rocket boosters (SRB's), which provide the main thrust for lift-off. System components are reusable except for the et. The system provides a large, usable space of 15 ft. In diameter and 60 ft in length in the orbiter cargo bay, with a payload delivery capability of 65,000 lb. (29,484 kg) to a 150 n. Mi. 28.50 circular orbit. The vehicle is approximately 184 ft. Long and weighs 4.5 million lb. At lift-off. The orbiter provides shirt-sleeve environment for the crew.

Arrow controls are used to page through windows.

Display-control options are clearly indicated.

Figure 6-1. Display Controls

6.1.2 Selection of Data for Display

Enable the user to select data for display. In determining the means for specification of data output, either by the user or automatically, consider the following recommendations:

6.1.2.1 For general data processing systems, enable users to specify the data for displayed output.

6.1.2.2 For specific information handling applications, always allow users to select data to meet task requirements.

6.1.2.3 If the designer cannot exactly determine the particular categories required by users at different stages of their job, permit users to select the categories they need for any information handling task.

6.1.2.4 Responses to simple requests for data display should not take more than 0.5 to 1.0 second.

6.1.2.4.1 If a response must be delayed, notify the user when display output is complete.

6.1.3 Update

When displayed data change dynamically with external events, permit the user to request an update of the change(s) and to control the update rate.

6.1.3.1 Consider highlighting data changes that result from automatic updating of a display.

6.1.3.1.1 Maintain the highlighting briefly but long enough to call the user's attention to any changes.

6.1.3.2 Ensure that changing data values are readable.

6.1.3.3 If the user must determine trends over time or make predictions based on changing data, provide integrated trend displays and predictive displays.

6.1.4 Suppression

Permit the user to suppress displayed data not required for the task at hand.

6.1.4.1 After data have been suppressed from a display, remind the user of the suppressed data.

6.1.4.2 Enable the user to restore suppressed data quickly to their originally-displayed form.

6.2 General Window Appearance

The appearance of *windows* in a given application is generally controlled by instructions built into the user-interface development software. In some cases, it may be possible to exercise options on window appearance or to customize some aspects of window appearance.

6.2.1 Window Components

Typically, a window is a square or rectangular area of the screen that contains a *window frame*, a *menubar*, an *application area*, and *window controls*. Optionally, a window may include a *command area* and *a message area*. The components of the window frame are sometimes called "decorations." Decorations may include a title bar, maximize and minimize buttons, a resize border, and a window menu button. A specific set of window controls may be required and built into the UI-development software. Other window controls are optional.

6.2.1.1. Display a brief, unique, and descriptive title at the top center of <u>each</u> window unless directed otherwise by a UI-toolkit style guide.

6.2.1.1.1 Continue to display the window title while the user scrolls data in the application area.

6.2.1.2 Tailor the size of the application area to the user's needs.

6.2.1.3 Provide window controls to support the following actions:

- Opening and closing a window.
- Moving a window.
- Re-sizing a window.
- Shrinking a window to an icon.

- Scrolling through window contents.
- Zooming in and out.

6.2.1.3.1 Provide standard buttons by which the user may control the size of the window (from minimum through variable to maximum).

6.2.1.3.2 Comply with the default appearance and placement of scroll bars for your environment. (Some windowing systems allow users to control placement of scroll bars.)

6.2.1.3.3 Display scroll bars in full contrast only for the active window (that is, the window that displays the user's current input)

6.2.1.3.4 Make optional window controls consistent in appearance with the required controls.

6.2.2 Primary and Secondary Windows

An application's *primary or main window* displays the essential data that the user needs to interact with the application. Additional, *secondary windows* convey context-specific information. Secondary windows may be called "transient" or "child" windows. A dialog box may be used to create a secondary window.

6.2.2.1 Limit the number of secondary windows to avoid creating navigation problems for the user.

6.2.2.2 Locate related window components in the same place from parent to child.

6.2.2.3 Use a secondary window to temporarily add data (such as help screens, menus, or other features) to a display, or as a means to control or display divergent information, or to segregate and control separate operations.

6.2.3 Tiled and Overlapping Windows

When windows are strictly tiled they abut on each other's borders but do not overlap. The look is two-dimensional. When windows overlap, the window nearest to the apparent foreground is the active window. It may obscure data "beneath" it or be made transparent so that underlying data are visible. The look of overlapping windows is close to three-dimensional, like that of papers stacked on a desk (Figure 6-2).

6.2.3.1 Use *tiled windows* to support user performance when any of the following apply:

- User activities focus on a single task.
- Tasks require minimal manipulation of windows.
- Users are novices or use the application infrequently.

6.2.3.2 Use *overlapping* windows when:

- User activities cut across independent tasks.
- Tasks require frequent window manipulation.
- Users are expert or use the application frequently.

6.2.3.3 Use a variant of tiled or overlapping windows to limit clutter and promote efficiency while supporting access to several sources of information. Variants include *non-space-filling tiling, piles of tiles, automatic panning, window zooming*, and *cascades*.

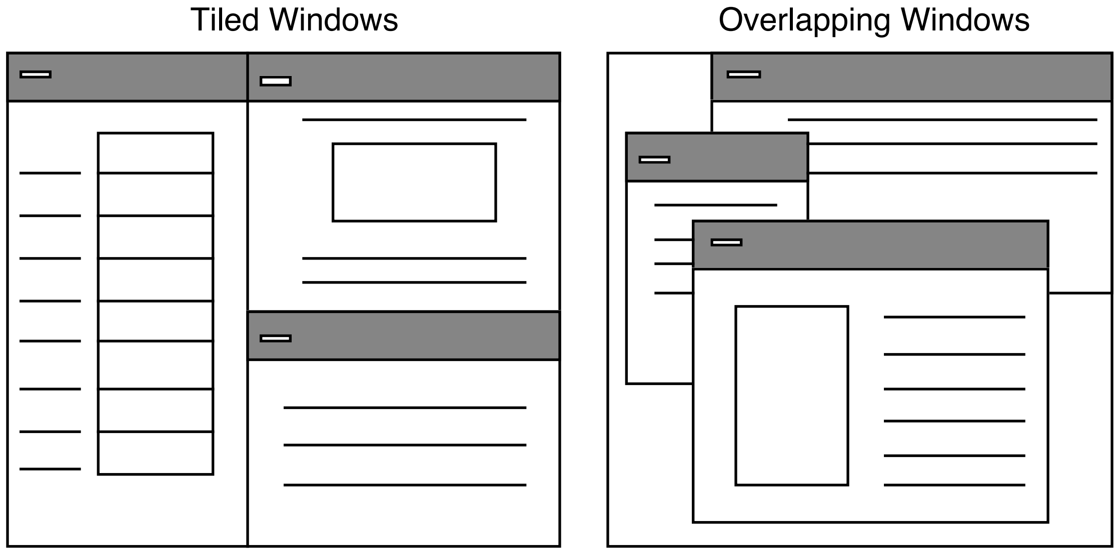


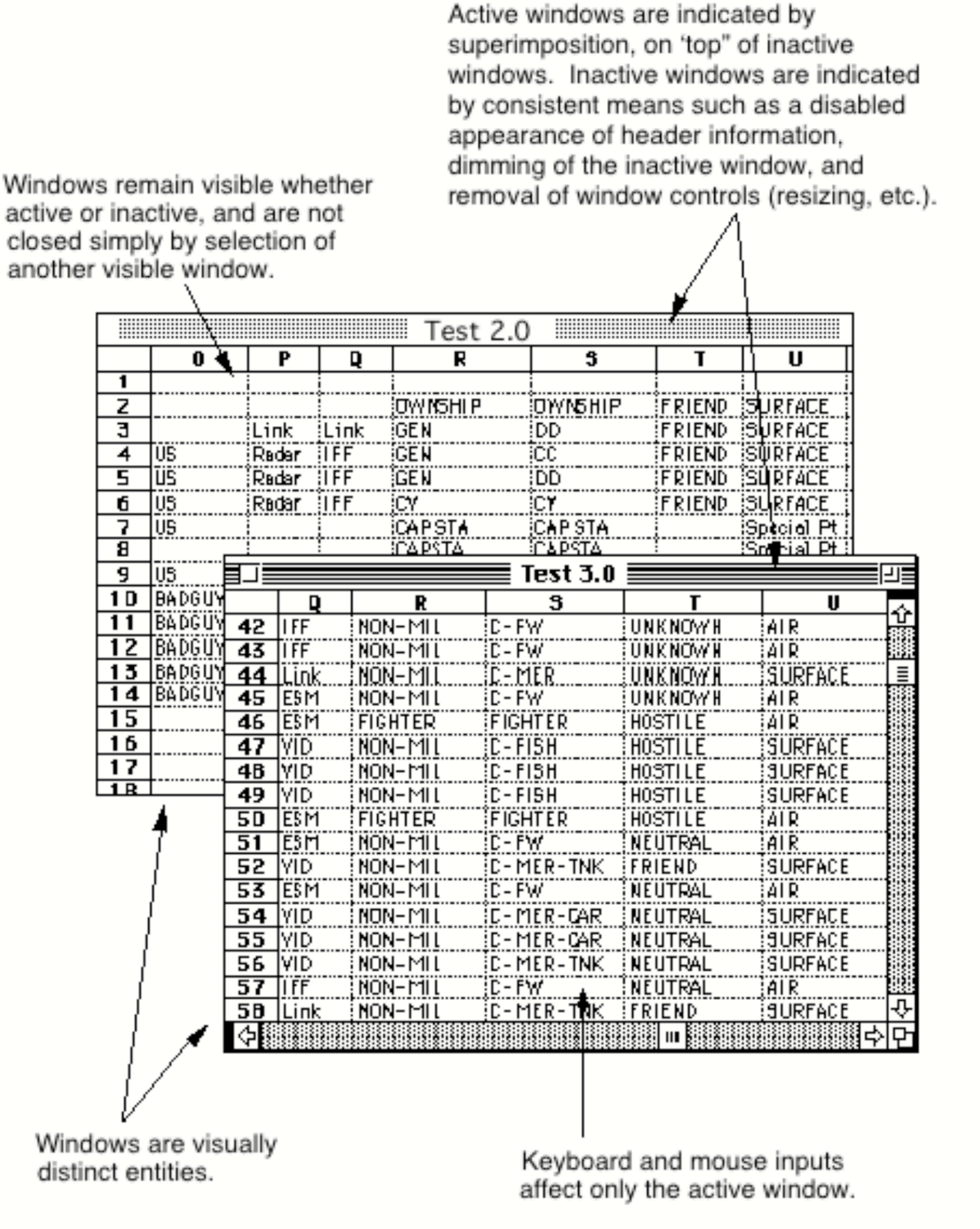
Figure 6-2. Tiled and Overlapping Windows

6.2.3.4 If using overlapping windows, ensure that window overlays are non-destructive; overlaid data are not permanently erased.

6.2.3.5 For tiled windows, indicate the active window by means of a technique such as highlighting the window border, altering the background color slightly, or changing the labeling.

6.2.3.6 For overlapping windows, indicate the active window by moving it to the forefront upon activation by the user (Figure 6-3).

6.2.3.7 For overlapping windows, use a neutral background pattern instead of complex patterns that create unintended visual effects.



Windows remain visible whether active or inactive, and are not closed simply by selection of another visible window.

Active windows are indicated by superimposition, on 'top" of inactive windows. Inactive windows are indicated by consistent means such as a disabled appearance of header information, dimming of the inactive window, and removal of window controls (resizing, etc.).

Test 2.0

	O	P	Q	R	S	T	U
1							
2				OWNSHIP	OWNSHIP	FRIEND	SURFACE
3		Link	Link	GEN	DD	FRIEND	SURFACE
4	US	Radar	IFF	GEN	CC	FRIEND	SURFACE
5	US	Radar	IFF	GEN	DD	FRIEND	SURFACE
6	US	Radar	IFF	CY	CY	FRIEND	SURFACE
7	US			CAPSTA	CAP STA		Special Pt
8				CAPSTA	CAPSTA		Special Pt
9	US						
10	BADGUY						
11	BADGUY						
12	BADGUY						
13	BADGUY						
14	BADGUY						
15							
16							
17							
18							

Test 3.0

	Q	R	S	T	U
42	IFF	NON-MIL	C-FW	UNKNOWN	AIR
43	IFF	NON-MIL	C-FW	UNKNOWN	AIR
44	Link	NON-MIL	C-MER	UNKNOWN	SURFACE
45	ESM	NON-MIL	C-FW	UNKNOWN	AIR
46	ESM	FIGHTER	FIGHTER	HOSTILE	AIR
47	VID	NON-MIL	C-FISH	HOSTILE	SURFACE
48	VID	NON-MIL	C-FISH	HOSTILE	SURFACE
49	VID	NON-MIL	C-FISH	HOSTILE	SURFACE
50	ESM	FIGHTER	FIGHTER	HOSTILE	AIR
51	ESM	NON-MIL	C-FW	NEUTRAL	AIR
52	VID	NON-MIL	C-MER-TNK	FRIEND	SURFACE
53	ESM	NON-MIL	C-FW	NEUTRAL	AIR
54	VID	NON-MIL	C-MER-CAR	NEUTRAL	SURFACE
55	VID	NON-MIL	C-MER-CAR	NEUTRAL	SURFACE
56	VID	NON-MIL	C-MER-TNK	NEUTRAL	SURFACE
57	IFF	NON-MIL	C-FW	NEUTRAL	AIR
58	Link	NON-MIL	C-MER-TNK	FRIEND	SURFACE

Windows are visually distinct entities.

Keyboard and mouse inputs affect only the active window.

Figure 6-3. Active and Inactive Windows

6.2.4 Decision-Supportive Window Design

The user's decision-making tasks will often require the integration of various kinds of information from multiple sources. Data may be given on different scales or in different units. From the perspective of the user's information-processing capacities, an objective of window design is to increase the user's ability to integrate such multiple sets of information. If the different sets of data must be kept separate and not integrated automatically, various windowing strategies can promote user productivity.

6.2.4.1 Consider using the suggested windowing strategies under the conditions specified as follows:

- When the user must consider different data sets in making a decision, place the required data sets in separate windows that can be displayed simultaneously.
- When it will benefit users to see the same data at different levels of analysis, display each level in a separate window.
- When the user can benefit from different perspectives on the same object or event, show each perspective in a different window. Perspectives might be more or less concrete or abstract, for example, looking at a spacecraft design from various angles, considering different attributes of the spacecraft, or taking an historical or predictive view of spacecraft operations.
- When changes made in one window affect data in other windows, link the affected windows so that the rest of the window set is updated by the changes.

6.2.4.2 Decide which of the decision-supportive strategies to use on the basis of a good understanding of the user's functions and tasks.

6.2.4.3 Use combinations of the decision-supportive strategies that will best support user decision making in particular situations.

6.2.4.4 If the user must interact with a sequence of displays in close temporal proximity, use separate windows that can be displayed simultaneously.

6.2.5 Related Versus Independent Windows

To promote the user's conceptual grouping, design related windows or sets of windows so that they share family characteristics. To promote differentiation of unrelated windows, windows or window sets that are independent of each other should reflect their differences partially through their appearance. (Independent windows may also behave differently.)

6.2.5.1 Use identical attributes for related windows (e.g., size, shape, color of foreground, background, and border).

6.2.5.1.1 Give independent windows contrasting attributes (e.g., different sizes or shapes, different colors in foreground, background, or borders).

6.2.5.2 Place related windows close to each other.

6.2.5.2.1 Use spacing to segregate independent windows from each other.

6.3 General Window Behavior ("Feel")

The behavior of tiled or overlapping windows may be entirely controlled by the windowing system, or some behaviors may be controlled by the user. For example, the user may control the basic windowing actions (e.g., open, resize, move), as well as placement of scroll bars. *Panning* strategies may be automatic or under user control. Although user-controlled placement of windows and window elements requires additional decision-making and housekeeping, some amount of optional user control may improve subjective satisfaction with the user interface.

General design objectives are to limit the extent of window manipulation and to preserve predictability in windowing behavior by (1) minimizing manual activity for window-manipulation tasks, and (2) aiming for consistency across windowing operations.

6.3.1 Window Manipulation

Systems vary in the steps required to perform window-manipulation actions, but the actions themselves have become relatively standardized. The behavior of controls located in the window frame is usually controlled by the windowing system in use.

6.3.1.1 Provide common window-manipulation actions, as listed in Table 6-1. These actions are typically performed with a mouse, but keyboard alternatives may be provided.

6.3.1.1.1 Following the user's open action, open the new window as close as possible to the *current focus* without obscuring the current focus. For example, when the user selects a control-panel icon, display the open control panel just below its icon. If the user requests help on a fill-in-form field, display the help window to the side of the field without obscuring the field.

6.3.1.1.2 Open and close windows as smoothly and rapidly as possible.

Table 6-1 Window Manipulation

OPENING AND CLOSING WINDOWS	
Create	Displays an entirely new window.
Delete	Removes a window from the screen.
Open	Replaces an iconic window with the full-size window it represents.
Close	Replaces a window with an iconic window.
Bring-To-Front	Moves a window to the most forward plane of a screen with overlapping windows.
Push-To-Back	Moves a window to the most rearward plane of a screen with overlapping windows.
CHANGING WINDOWS	
Move	Repositions a window in its two-dimensional plane.
Resize	Shows more or less of the data in a window by contracting the window or expanding it to its maximum size.
Zoom	Expands to maximum size with one action.
Rescale	Shows more or less of the data in the window by changing the scale of the image in the window.
Scroll	Selects a different portion of data for viewing without resizing.
Name/Rename	Defines or changes the name of a window.
Make Active (Change Focus)	Designates a window as the one available for interaction.

(Source: Billingsley, 1988)

6.3.1.1.3 Upon resizing, reformat text, graphics, or icon layouts so that they remain visible, up to some standard limit set by project consensus.

6.3.1.1.4 Protect against any obscuring of critical information during window resizing.

6.3.1.1.5 Provide visual feedback during movement of a window.

6.3.1.1.6 Follow a standard panning sequence.

6.3.1.1.7 If the user must frequently perform long mouse movements, provide keyboard alternatives for windowing actions.

6.3.1.1.8 Consider controlling window manipulation by user task actions, without requiring the user to perform direct window-manipulation actions.

6.3.1.2 Allow keyboard and mouse input to affect only the active window (Figure 6-3).

6.3.2 Aiding Window Arrangement and Navigation

To reduce housekeeping and support the user's sense of orientation, provide some built-in aids to window manipulation and navigation between windows.

6.3.2.1 Design for maximum ease and efficiency of navigating between windows.

6.3.2.2 Consider providing a way for the user to move around among open windows with minimal manual activity.

6.3.2.3 Consider providing a capability for the user to call by name a window or a particular configuration of windows (a *window macro*).

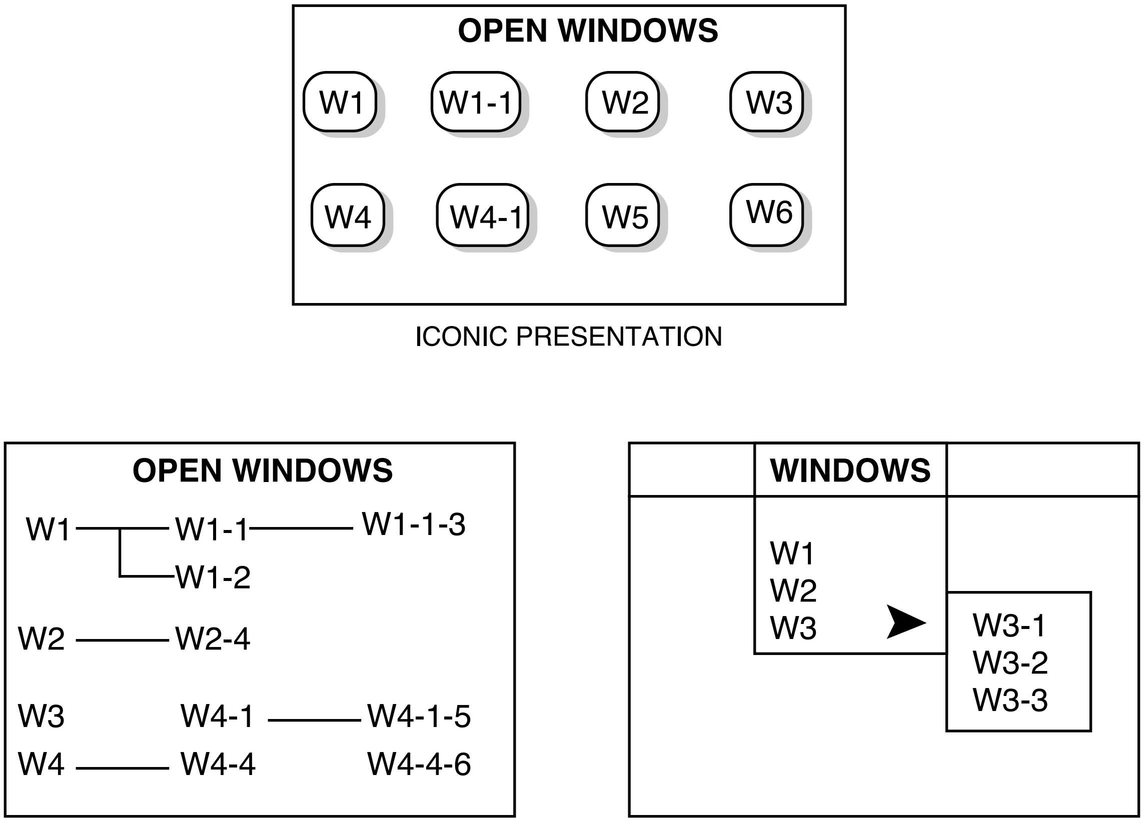
6.3.2.4 When windows are allowed to overlap, provide powerful commands to support the user's tailored arrangement of windows on the screen (e.g., create, name, and recall specific sets of windows).

6.3.2.5 Within a set of windows (parent and children), offer an indication of how far the user has progressed through the set and how many windows remain in the set.

6.3.2.6 For a set of overlapping, open windows, allow the user to request an iconic or graphical map depicting all the open windows (Figure 6-4).

6.3.2.7 When a map of open windows is available, allow the user to designate the active window by selecting a map element. Bring the selected window forward without requiring the user to resize or move other windows.

6.3.2.8 Consider opening/closing child windows when a parent window is opened or closed.



ICONIC PRESENTATION

FLOWCHART PRESENTATION

PULL DOWN WINDOW PRESENTATION

Figure 6-4. Iconic or Graphical Map of Open Windows

(Source: Avery and Bowser, 1992)

References

6.0	Billingsley (1988); Shneiderman (1992)
6.1	Carlow (1992); Smith and Mosier (1986)
6.1.1	Smith and Mosier (1986)
6.1.1.1	Carlow (1992)
6.1.1.2	Smith and Mosier (1986)
6.1.1.3	Carlow (1992); Smith and Mosier (1986)
6.1.1.4	Smith and Mosier (1986)
6.1.1.3.1	Carlow (1992)
6.1.1.3.2	Carlow (1992); Smith and Mosier (1986)
6.1.1.3.3	Carlow (1992)
6.1.2	Smith and Mosier (1986)
6.1.2.1	Smith and Mosier (1986)
6.1.2.3	Smith and Mosier (1986)
6.1.2.4	Smith and Mosier (1986)

6.1.2.4.1	Smith and Mosier (1986)`
6.1.3	Smith and Mosier (1986)
6.1.3.1	Smith and Mosier (1986)
6.1.3.1.1	Smith and Mosier (1986)
6.1.3.2	Smith and Mosier (1986)
6.1.3.3	Smith and Mosier (1986); Thurman (1995); Thurman and Mitchell (1995)
6.1.4	Smith and Mosier (1986)
6.1.4.1	Smith and Mosier (1986)
6.1.4.2	Smith and Mosier (1986)
6.2	e.g., Apple Computer (1987); OSF (1991)
6.2.1	Apple Computer (1987); OSF (1991); Shneiderman (1992)
6.2.1.1	Fowler and Stanwick (1995); Shneiderman (1992); Smith and Mosier (1986)
6.2.1.1.1	Carlow (1992)
6.2.1.2	Carlow (1992)
6.2.1.3	Apple Computer (1987); Billingsley (1988); Carlow (1992); Galitz (1993); Mayhew (1992); OSF (1991); Shneiderman (1992)
6.2.1.3.1	e. g., Apple Computer (1987); OSF (1991)
6.2.1.3.2	e.g., Apple Computer (1987); OSF (1991)
6.2.1.3.3	Galitz (1992)
6.2.1.3.4	Galitz (1992)
6.2.2	Apple Computer (1987); OSF (1991)
6.2.2.1	Mayhew (1992)
6.2.2.2	Shneiderman (1992)
6.2.2.3	Carlow (1992); Smith and Mosier (1986)
6.2.3	Galitz (1993); Shneiderman (1992)
6.2.3.1	Galitz (1993)
6.2.3.2	Galitz (1993)
6.2.3.3	Shneiderman (1992)
6.2.3.4	Smith and Mosier (1986)
6.2.3.5	Galitz (1993); Shneiderman (1992); Smith and Mosier (1986)
6.2.3.6	Carlow (1992); Mayhew (1992); Shneiderman (1992)
6.2.3.7	Tufte (1990)
6.2.4	Mitchell and Saisi (1987); Shneiderman (1992)
6.2.4.1	Mayhew (1992)
6.2.4.2	Shneiderman (1992)
6.2.4.3	Shneiderman (1992)
6.2.4.4	Carlow (1992)
6.2.5	Galitz (1992)
6.2.5.1	Shneiderman (1992)
6.2.5.1.1	Shneiderman (1992)
6.2.5.2	Tullis (1988)
6.2.5.2.1	Tullis (1988)
6.3	Mayhew (1992); Shneiderman (1992)
6.3.1	Billingsley (1988); Apple Computer (1987); OSF (1991)
6.3.1.1	Billingsley (1988)

6.3.1.1.1	Shneiderman (1992)
6.3.1.1.2	Shneiderman (1992)
6.3.1.1.3	Shneiderman (1992)
6.3.1.1.4	Shneiderman (1992)
6.3.1.1.5	Shneiderman (1992)
6.3.1.1.6	Shneiderman (1992)
6.3.1.1.7	Galitz (1993); Shneiderman (1992)
6.3.1.1.8	Shneiderman (1992)
6.3.1.2	Carlow (1992)
6.3.2	Mayhew (1992); Shneiderman (1992)
6.3.2.1	Mayhew (1992)
6.3.2.2	Mayhew (1992)
6.3.2.3	Mayhew (1992)
6.3.2.4	Mayhew (1992)
6.3.2.5	Mayhew (1992)
6.3.2.6	Avery and Bowser (1992)
6.3.2.7	Avery and Bowser (1992)
6.3.2.8	Shneiderman (1992)

7.0 Guidelines For Visual-Coding Techniques

Color is a major technique for supporting the user's quick detection and discrimination of displayed data elements or groupings of data. Other visual-coding techniques include variations in brightness, flashing, line style, symbols, sizes, and shapes. Each of these techniques can be misused or overused. A rule of thumb is to use visual coding for functional, not decorative purposes. The key challenge to designers is to use these techniques in ways that enhance users' task performance (Table 7-1).

Table 7-1 Attention-Getting Techniques

Technique	When To Use	When Not To Use
Color	A powerful attention getter is needed.	Several colors are already used on the display.
Blinking/Flashing	User must respond immediately.	Message does not require immediate attention or text is lengthy.
Bold	Captions or titles should stand out.	There are more than three levels of bold.
Reverse Video	Purpose is to indicate a selected item or error.	Text is lengthy.
Size	A code is needed for relative quantity or importance.	There are 5 or more levels of size codes.
Font	Text items should stand out.	There are more than 2 to 4 different character types.
Underlining	Purpose is to draw attention to key words for instruction or to distinguish fill-in fields from text.	Too frequent use reduces legibility.
Shape	Purpose is to communicate urgency.	Use of shapes is not based on established standards.
Special Characters and Icons	Purpose is to draw attention to items on a screen.	Space is limited (graphic symbols take up additional screen space).
Proximity	White space can be used to associate items with each other (grouping).	There is no need to associate items.
Borders	Purpose is to identify meaningful groups, create sense of closure, and/or focus attention.	Identified groups, closure, and focus of attention are not required.

(Source: Mayhew, 1992)

7.1 Color

Color is used for several functional purposes in screen design:

- To establish relationships between displayed objects.

- To help the user distinguish between displayed objects.

- To communicate the organization of information.

- To call the user's attention to system states.

If color is overused, these purposes are defeated. Likewise, if color is used for decorative purposes, these functional purposes are compromised.

7.1.1 Number of Colors

Limit the number of colors to be used. Preferably, use four or less.

7.1.1.1 Use task performance requirements as the basis for determining the number of colors to be presented simultaneously. In general, minimize the number of colors presented together on the same screen.

7.1.1.1.1 Use color to support visual-search tasks and symbol-identification tasks.

7.1.1.2 Use no more than six distinct colors or three shades of gray if the user must recall the meanings of colors or shades.

7.1.1.3 Use no more than six distinct colors if the user must perform rapid visual searching based on color discrimination.

7.1.1.4 If functional requirements dictate the use of more than the recommended number of colors or shades of gray, display a legend of color/shade meanings.

7.1.2 Pairing Colors

Avoid the color combinations listed in Table 7-2.

Table 7-2 Color Combinations to Avoid

Saturated Red and Blue	Saturated Red and Green
Saturated Blue and Green	Saturated Yellow and Green
Yellow on Purple	Green on White
Yellow on Green	Blue on Black
Magenta on Green	Red on Black
Magenta on Black	Yellow on White

7.1.3 Foreground and Background Colors

Consult Table 7-3 for recommended foreground and background colors.

Use a medium achromatic background (e.g., dark or medium gray) to maximize the visibility of foreground colors.

7.1.3.1 Consider the effects of varying levels of saturation (color intensity) and the effects of varying levels of lightness (amount of white mixed with the color) on the discriminability of colors and on color interactions (Figure□-2).

7.1.3.1.1 Avoid combinations that are similar in lightness (e.g., navy blue on black, yellow on white).

7.1.3.2 Maintain adequate contrast between foreground and background colors to enhance color perception and perceived image resolution.

7.1.3.2.1 Increase contrast if the screen will be viewed under dim lighting conditions.

7.1.3.2.2 Consider using complementary colors (yellow on dark blue; magenta on green) to maximize color contrast, if appropriate for the user's task environment.

7.1.3.3 Consult authoritative sources for guidance on measuring color uniformity, color contrast, and color differences.

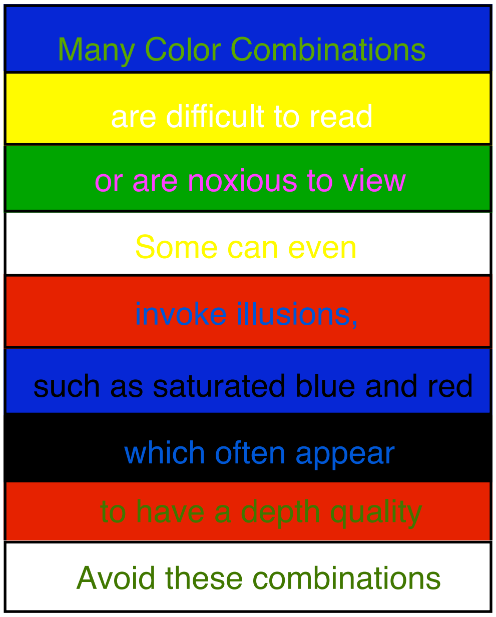
7.1.3.4 Test selected colors with users to verify that the colors can be easily discriminated from each other. Color contrast should be at least 7:1 for foreground versus background

Table 7-3. Foreground/Background Colors

Background

Foreground	Black	Blue	Green	Cyan	Red	Magenta	Brown	White	Gray	Beige
Black	-------			R		R		R	R	R
Blue	-------	------	------	------	-----	------	------	r	r	r
Green	-------	r	------	R	-----				-----	-----
Cyan	R			------					R	R
White	R								-----	-----
Bold Green	R								R	R
Bold Cyan	R	R		------	R	R			R	R
Bold Magenta	R		R			------			R	R
Yellow	R	R	------	R		R		-----	-----	-----
Bold white	R		R	R				-----	-----	-----

(R = recommended; r = some saturations acceptable; ------ = avoid this combination)



Many Color Combinations

are difficult to read

or are noxious to view

Some can even

invoke illusions,

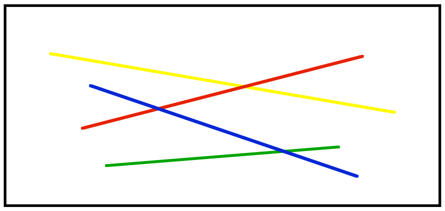
such as saturated blue and red

which often appear

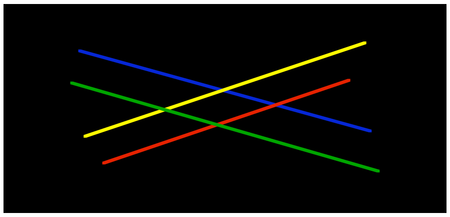
to have a depth quality

Avoid these combinations

On a white background, colors such as red, blue and green are visible. Yellow is dificult to perceive because of its low contrast with white.



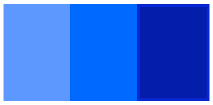
On a black background, colors such as red and yellow are visible, while blue and green do not have adequate contrast with black.



Light blue on white background can be difficult to read, since the human eye occasionally does not focus blue very well.

Medium blue on a white background is easier to read, but still may be inadequate.

Dark blue, however, contrasts adequately with white.



The strongest contrast is between black and white

Humans can discriminate thousands of colors, but generally recognize only about 5-12 (family of blue, family of green, etc.) Don’t overuse color.

Try to match all the colors in the palette below, without benefit of direct comparison. Can you readily associate every color when distributed?

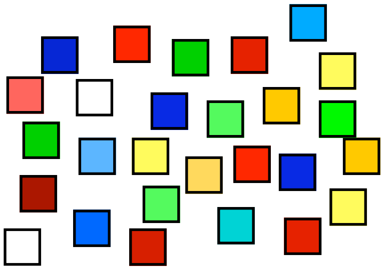


Figure 7-1. Use of Color

7.1.4 Colors For Thin Lines

Consult Table 7-4 for colors to make thin lines easily perceivable.

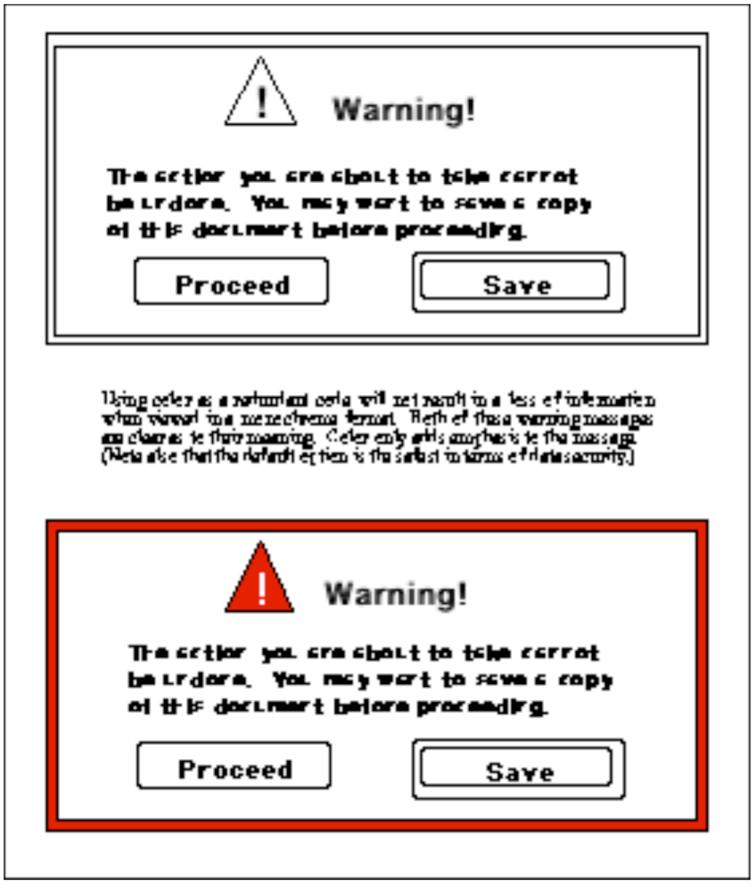
Table 7-4 Colors to Use for Thin Lines

# of Colors	White Background	Black Background
1	Red or Green	Yellow, Cyan or Green
2	Red and Green Magenta and Cyan Red and Blue	Green and Magenta Yellow and Magenta Cyan and Magenta
3	Red, Blue and Green	Cyan, Magenta and Yellow

7.1.5 Redundant Use of Color

Be sure that the information coded by color is coded in some other manner for the sake of color-deficient users (Figure 7-2).

7.1.5.1 Before adding color to a display, design for its use in monochrome. Then, use color as a redundant code to enhance an otherwise logical, well-organized design.



The action you are about to take cannot be undone. You may want to save a copy of this document before proceeding.

Using color as a redundant code will not result in a loss of information when viewed in a monochrome format. Both of these warning messages are clear as to their meaning. Color only adds emphasis to the message. (Note also that the default option is the safest in terms of data security.)

The action you are about to take cannot be undone. You may want to save a copy of this document before proceeding.

Figure 7-2. Redundant Color Coding

7.1.6 Consistent Use of Color

Use color consistently with common meanings for the user's culture (e.g., in Western cultures, use green for good or normal, yellow for caution, and red for danger or warning).

7.1.6.1 If the user community has previously established meanings for various colors, retain those meanings. Do not use a color to signify a different condition than it signified in the previous system.

7.2 Brightness

On either monochrome or color displays, variations in brightness can be used to attract the user's attention and to differentiate between categories or levels within a category. On a color display, variations on the continuum from dull to bright (brightness) are indistinguishable from variations in lightness (the range from white through gray to black).

7.2.1 Suggested Uses

Use brightness coding for task-related purposes.

7.2.1.1 Consider using high brightness to call attention to errors in data-entry fields and to highlight answer fields on question-and-answer screens.

7.2.1.2 Consider using brightness coding to differentiate between adjacent items of information or to code state conditions (e.g., on/off, standby-run).

7.2.1.3 Consider using "reverse video" (i.e., brightness inversion) to highlight critical items requiring user attention.

7.2.1.3.1 Return to a normal brightness when the user has responded.

7.2.2 Levels of Brightness

Use only two levels of brightness coding (bright and dim) separated by at least a 2:1 ratio.

7.3 Flashing/Blinking

Flashing or blinking is a powerful attention getter but should be used only rarely. Overuse of flash or blink coding has a high potential to distract the user. Data or text that the user must read should never blink or flash because a blinking object is, by definition, not displayed continuously and can be read only when it is displayed.

7.3.1 Suggested Uses

Use flash or blink coding only to indicate an urgent need for user attention and response or to indicate the active location for data entry (i.e., place a blinking cursor at the point where user input will be accepted).

7.3.1.1 Allow only a very small area of the screen to flash at any moment.

7.3.1.2 Instead of blinking the data or text, place a small square or rectangle next to the critical information, and blink just that shape.

7.3.2 Levels of Flashing

Use no more than two levels of flash/blink coding:

- Slow = less than 2 Hz
- Fast = 3 to 5 Hz

7.3.3 Length of Intervals

In general, use equal "on" and "off" duty cycles.

7.3.3.1 Consider "winking" rather than blinking by using an "on" cycle that is substantially longer than the "off" cycle.

7.4 Line Coding

Lines or "rules" aid in focusing the user's attention on related information and serve to separate unrelated groupings of information. Line borders delineate the boundaries of menubars, display-control options, and entire windows. Line coding should be used sparingly.

7.4.1 Line Attributes

Use line thickness, width, and height or length to convey functional meaning.

7.4.1.1 Use line coding by type (e.g., solid, dashed, dotted), by width, or by other attributes to indicate association between elements.

7.4.1.1.1 Limit to four the number of variations in any one line attribute (e.g., four thicknesses, four lengths).

7.4.1.1.2 Minimize the possible interactions caused by varying line attributes. For example, use horizontal lines of equal width and vertical lines of equal height where possible.

7.4.1.2 Use line-length coding for applications involving spatial categorizations in a single dimension (e.g., velocity).

7.4.1.3 Use line-direction coding for applications involving spatial categorizations in two dimensions (e.g., altitude, bearing).

7.5 Special Symbols

Use special, standard symbols (e.g., asterisk, arrow) to draw the user's attention to specific items in alphanumeric displays.

7.5.1 Mappings

Make special symbols analogous to the event or system elements they represent, based on established standards or conventional meanings.

7.5.2 Consistent Meaning

Assign consistent meanings to special symbols within and across applications.

7.6 Symbol Sizes

Use size coding only on uncrowded displays.

7.6.1 Number of Sizes

Use no more than three symbol sizes for coding. A larger sized object should be 1.5 times the height of the next smaller object (e.g., character, symbol, shape).

7.7 Shapes

Use shape categories (e.g., circle, triangle, square) to code related objects and to support the user's ability to discriminate between various categories of displayed data.

7.7.1 Redundant Coding

Map shapes to colors (e.g., red circles, green triangles) to achieve redundant coding (i.e., meaning is conveyed both by shape and color).

7.7.2 Distinctiveness

Use a distinctive shape to call attention to a single type of displayed object.

7.8 Type Styles

Variations in type styles include the use of bold, underlining, and different fonts.

7.8.1 Bold

Use the heavy intensity of bolding for strong emphasis.

7.8.1.1 If more than one level of bold is available, avoid using more than three different levels whose meanings the user must remember.

7.8.2 Underlining

Use underlining for mild emphasis, not for urgent or critical information.

7.8.2.1 Use underlining to emphasize key words, titles or headings.

7.8.2.2 Use underlining to distinguish fill-in fields from surrounding text or labels.

7.8.3 Fonts

Use multiple fonts to convey moderate emphasis and to help the user discriminate among categories of displayed information.

7.8.3.1 Limit to two or three the number of different fonts displayed on any one screen. Do not exceed four fonts.

7.8.3.2 Use variations within one font (e.g., size, style) to convey levels within a category of information. For example, large, bold letters for high-level titles; smaller, bold letters for second-level titles; still smaller, underlined letters for third-level titles).

7.9 Three-Dimensional Effects

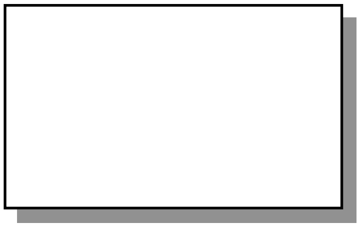
The GUI toolkits typically provide widgets with three-dimensional features, such as drop shadows and beveled edges. Three-dimensional features cause objects to appear closer to the user. Closer objects (i.e., those in the foreground) take on greater importance than background elements. The user's attention goes naturally to screen elements that seem closer because they have become important in the context of the task. Such features should be used sparingly on any one display screen to avoid canceling out their effects.

7.9.1 Drop Shadows

Consider using drop shadows to make important elements appear closer to the user.

7.9.1.1 Place shadows at the lower right of icons and buttons, as if light were coming from the upper left.

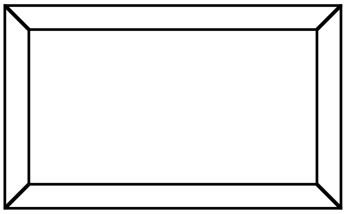
7.9.1.2 Place shadows along the bottom and right side of a pull-down menu, dialog box, or window to attract the user's attention.



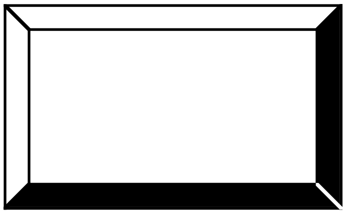
Drop Shadow

7.9.2 Beveled Edges

Consider using beveled edges to bring important screen elements into the foreground. Beveled edges may be used on icons, buttons, menus, dialog boxes, and windows.



Plain Beveled Edges



Beveled Edges
with Shading

7.9.2.1 Shade the bottom and right beveled edges to enhance the three-dimensional effect.

References

7.0	Mayhew (1992); Shneiderman (1992); Smith and Mosier (1986)
7.1	Mayhew (1992); Shneiderman (1992)
7.1.1	Fowler and Stanwick (1995); ISO (1993, Part 8); Marcus (1992); Mayhew (1992); Shneiderman (1992); Smith and Mosier (1986)
7.1.1.1	ISO (1993, Part 8); Mayhew (1992); Shneiderman (1992)
7.1.1.1.1	Mayhew (1992); Christ (1975)
7.1.1.2	ISO (1993, Part 8)
7.1.1.3	ISO (1993, Part 8)
7.1.1.4	ISO (1993, Part 8); Mayhew (1992)
7.1.2	Carlow (1992)
7.1.3	Carlow (1992); ISO (1993, Part 8); Mayhew (1992)
7.1.3.1	Thorell and Smith (1990)
7.1.3.1.1	Horton (1990)
7.1.3.2	Thorell and Smith (1990)
7.1.3.2.1	Travis (1991)
7.1.3.2.2	Horton (1991); Thorell and Smith (1990)
7.1.3.3	Carter and Carter (1982); ISO (1993, Part 8); Murch (1987); Silverstein (1987); Thorell and Smith (1990); Travis (1991); Wyszecki and Stiles (1982)
7.1.3.4	Mayhew (1992); Smith and Mosier (1986); Tufte (1990)
7.1.4	Carlow (1992); Mayhew (1992)
7.1.5	Carlow (1992); Marcus (1992); Mayhew (1992); Smith and Mosier (1986)
7.1.5.1	Fowler and Stanwick (1995); Mayhew (1992); Shneiderman (1992); Smith and Mosier (1986); Travis (1991)
7.1.6	Marcus (1992); Shneiderman (1992)
7.1.6.1	Shneiderman (1992)
7.2	Mayhew (1992)
7.2.1	Mayhew (1992)
7.2.1.1	Galitz (1993)
7.2.1.2	Carlow (1992)
7.2.1.3	Carlow (1992); Galitz (1993); Mayhew (1992); Smith and Mosier (1986)
7.2.1.3.1	Carlow (1992)
7.2.2	Galitz (1993); Carlow (1992); DoD (1989)
7.3	Galitz (1993); Mayhew (1992)
7.3.1	Galitz (1993); Mayhew (1992); Carlow (1992)
7.3.1.1	Carlow (1992)
7.3.1.2	Brown (1988)
7.3.2	Galitz (1993); Carlow (1992); DoD (1989)
7.3.3	Galitz (1993); ANSI/HFS (1998)
7.3.3.1	Galitz (1993); Smith and Mosier (1986); Carlow (1992)
7.4	Galitz (1993); Smith and Mosier (1986)
7.4.1	Galitz (1993); Horton (1991)
7.4.1.1	Horton (1991); Carlow (1992)

7.4.1.1.2	Galitz (1993); Tufte (1990)
7.4.1.2	Smith and Mosier (1986); Carlow (1992)
7.4.1.3	Smith and Mosier (1986); Carlow (1992)
7.5	Mayhew (1992); Smith and Mosier (1986); Carlow (1992)
7.5.1	Smith and Mosier (1986); Carlow (1992)
7.5.2	Smith and Mosier (1986)
7.6	Smith and Mosier (1986)
7.6.1	Smith and Mosier (1986); Carlow (1992)
7.7	Mayhew (1992); Smith and Mosier (1986)
7.7.1	Smith and Mosier (1986)
7.7.2	Mayhew (1992)
7.8	Galitz (1993); Mayhew (1992)
7.8.1	Galitz (1993); Mayhew (1992)
7.8.1.1	Mayhew (1992)
7.8.2	Galitz (1993)
7.8.2.1	Galitz (1993); Mayhew (1992)
7.8.2.2	Mayhew (1992)
7.8.3	Galitz (1993)
7.8.3.1	Mayhew (1992)
7.8.3.2	Galitz (1993)
7.9	Galitz (1993); Marcus (1992)
7.9.1	Galitz (1993)
7.9.1.1	Fowler and Stanwick (1995); Galitz (1993); Marcus (1992)
7.9.1.2	Galitz (1993); Marcus (1992)
7.9.2	Galitz (1993)
7.9.2.1	Galitz (1993); Marcus (1992)

8.0 Guidelines for User Guidance and Feedback

Providing effective guidance and feedback to the user contributes to meeting several general objectives of UI design:

- Consistency of operational procedures.
- Efficient use of system capabilities.
- Limited memory load on the user.
- Reduced learning time.
- Flexibility in supporting different classes of users.
- Error prevention.

User guidance includes prompts, warnings, error messages, status information, and on-line help. A key design goal is to reflect the user's (not the designer's) understanding of the system. Meeting this goal requires review and usability testing by potential end users who have not been involved with system development.

8.1 Prompts and General Guidance Messages

Prompts are advisory messages that help in navigating through the user interface. General guidance messages inform the user about available actions and suggest actions to be taken by the user.

8.1.1 Distinctive, Consistent Prompts and Messages

Ensure that prompts and general guidance messages are distinct from displayed data and used consistently throughout the application (Table 8-1).

8.1.1.1 Use consistent, concise phrasing, and minimal, consistent punctuation in prompts and general guidance messages.

8.1.1.2 Use positive, clear wording and the active voice in prompts and general guidance messages.

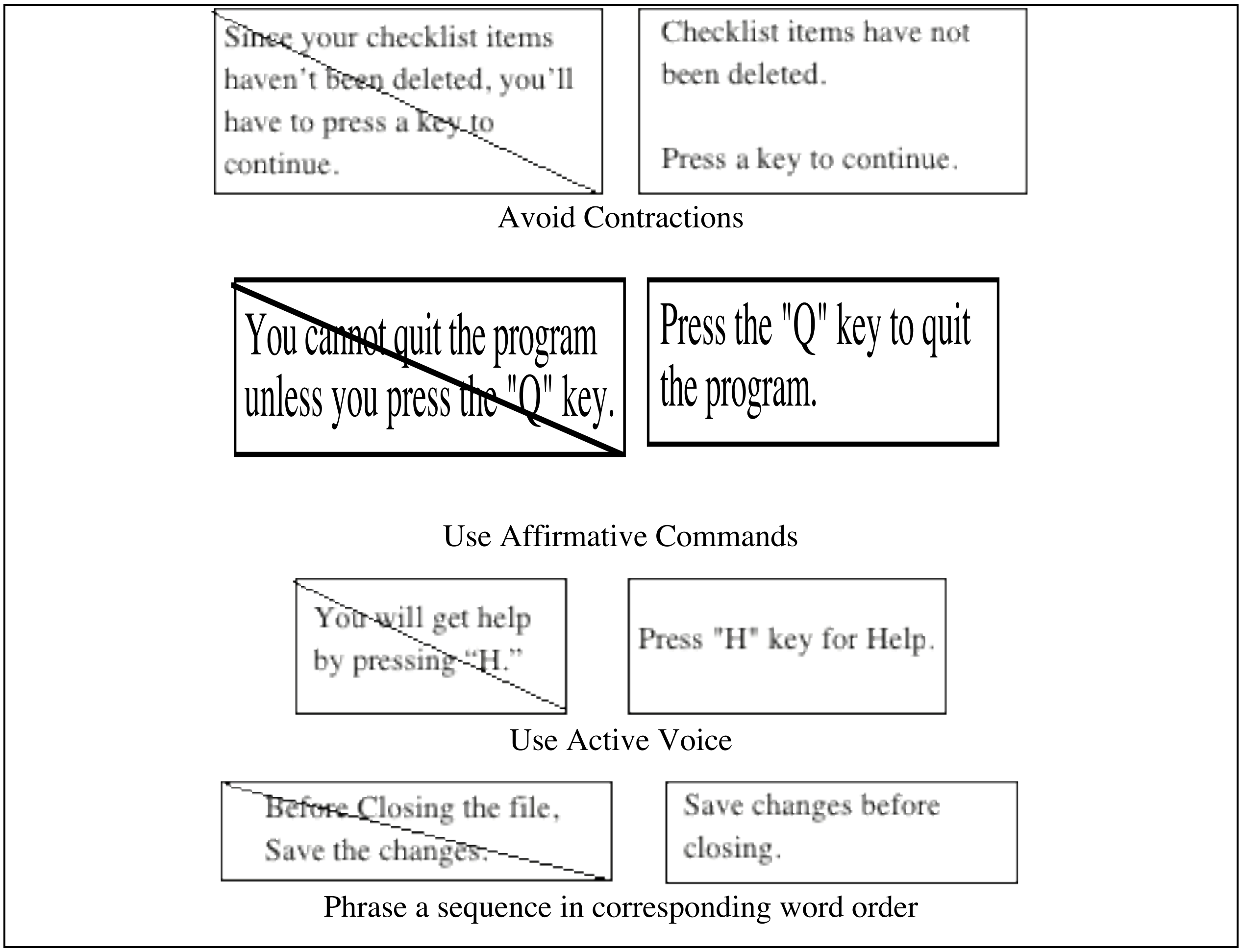
8.1.1.3 Make prompts and user guidance explicit to eliminate the need for the user to memorize lengthy sequences or refer to references.

8.1.1.3.1 Do not include codes or references to external sources in prompts.

8.1.1.4 Do not present prompts and general guidance in a person-like or personal manner. Do not use the first ("I") or second person ("You").

8.1.1.5 Where possible, provide prompts for required formats and acceptable values for data entry.

Table 8-1. Phrasing Prompts and General Guidance



~~Since your checklist items haven't been deleted, you'll have to press a key to continue.~~

Checklist items have not been deleted.

Press a key to continue.

Avoid Contractions

~~You cannot quit the program unless you press the "Q" key.~~

Press the "Q" key to quit the program.

Use Affirmative Commands

~~You will get help by pressing "H."~~

Press "H" key for Help.

Use Active Voice

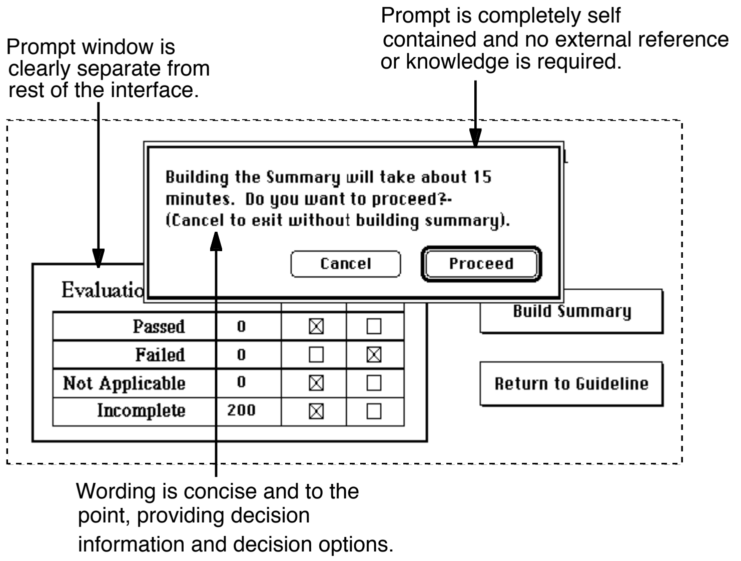
~~Before Closing the file, Save the changes.~~

Save changes before closing.

Phrase a sequence in corresponding word order

8.1.2 Explanations

Use brief prompts and messages to explain required input, commands, error messages, system capabilities, display formats, procedures, and steps in a sequence (Figure 8-1).



Prompt window is clearly separate from rest of the interface.

Prompt is completely self contained and no external reference or knowledge is required.

Building the Summary will take about 15 minutes. Do you want to proceed?- (Cancel to exit without building summary).

Passed	0	☒	□
Failed	0	□	☒
Not Applicable	0	☒	□
Incomplete	200	☒	□

Wording is concise and to the point, providing decision information and decision options.

Figure 8-1. Characteristics of Prompts

8.1.3 Prompting for Coded Data Entry

Prompt the user for required formats and provide values for valid data entries. For example, prompt the user to enter a color selection:

Color: ___
r = red
b = blue
g = green

8.1.4 Optional Guidance

Permit users to request prompts and general guidance, as needed, depending on their level of experience.

8.1.4.1 Design prompts and general guidance with the novice user in mind.

8.1.4.2 Allow expert users to "turn off" prompting and guidance.

8.1.5 Location of Prompts

Locate prompts where the user is expected to input data, whenever possible; otherwise place prompts in a standard message area.

8.2 Cautions and Warnings

Display of cautions and warnings should be consistent throughout the system.

8.2.1 Cautions

Use cautions for operations/actions that might have the following kinds of results:

- Permanent data loss.
- No data loss, but undoable consequences.
- Conflict with the operations of others.
- Need for excessive processing time.

8.2.2 Warnings

Use warnings for operations/actions that might have the following kinds of results:

- Permanent data loss.
- Data loss and undoable, with some exceptions (e.g., saving an edited file using the same file name as the edited file does not require a warning).
- Threat/compromise of information, such as proprietary information.
- Non-interruptable processing.
- Invocation of ancillary system actions (e.g., automatic deletion or overwriting of an original file when executing a renaming command).

8.2.3 Visual and Auditory Display

Present distinctive cautions and warnings through both visual and auditory display.

8.3 Error Messages

Use error messages to inform the user of incorrect command or data entry.

8.3.1 Location

Present the error message at the point of the error or in a consistently located message area.

8.3.2 Style

Make error messages specific, informative, and brief.

8.3.2.1 Use neutral wording in error messages. Keep the phrasing positive and professional.

8.3.2.2 Phrase error messages in terms of the current task.

8.3.3 Information Content

Include the following information in the error message:

- Why input was rejected.
- What corrective action(s) may be taken in subsequent operations.
- Format requirements, if formatting conventions were violated.

8.3.4 Detailed Explanation of Error

Permit the user to request a more detailed explanation of the error and additional information about the on-going operation.

8.3.5 Error Correction

Allow the user to correct an error immediately after it has been detected.

8.3.5.1 If an error cannot be corrected, design the system so that other transactions can still be initiated. Permit the user to store and later retrieve the transaction with an error.

8.4 System and Status Information

System/status messages should not be humorous or sarcastic. Nor should they be presented in the first ("I") or second person ("You").

8.4.1 Message Scope and Content

Display messages to indicate the following conditions:

- Keyboard Lock
- Log-on Denial Failure
- File Writing Operations
- Making Remote Connections
- Printing Progress/Spooling
- Complex Time Consuming Operations
- Processing Delay
- Save Operations
- Mail/Data Transmission
- High Time Shared System Loads
- Complex Processing Completion

8.4.1.1 Include the following information:

- A description of the system state.
- Directives for user action.
- Consequences, if any, of failure to follow directives.

8.4.2 Message Location

Present system/status messages in a consistent location.

8.4.3 Operational Mode

Inform the user of the current operational mode when the mode might affect the user's actions.

8.5 Task-Related Job Aids

Job aids support user productivity and performance by providing task-related guidance.

8.5.1 Content

Job aids should include an on-line dictionary of abbreviations, acronyms, and codes, as well as allowable options and value ranges. On-line help is a major job aid.

8.5.2 Dialog Aiding

Provide context-specific information on semantics and syntax of any available user dialog. Include a structured listing of the following command information:

- Each command available.
- The associated menu options and keystroke alternatives (accelerators).
- Uses and consequences of the command.

8.5.3 On-Line Help

Provide context-sensitive, self-explanatory help that permits rapid access to information about specific interactions, tasks, messages, or commands. Include help on questions users are likely to have about all parts of the application.

8.5.3.1 While the user is viewing help in a separate, reserved window, display as much as possible of the user's current task display.

8.5.3.2 If the user is executing a particular command and requests help, provide information on that command (e.g., the effects of that command; alternatives to that command).

8.5.3.3 When an error occurs, and the user requests help, provide a useful description of the error and suggest at least one recovery technique.

8.5.3.4 If an on-line reference manual is available, open the manual at the topic corresponding to the current context when the user requests help.

8.5.3.5 Use a help icon on the screen and designate a function key as the help key. Use the help icon and the help key consistently throughout the application.

References

8.0	Smith and Mosier (1986)
8.1	Smith and Mosier (1986)
8.1.1	Carlow (1992); Smith and Mosier (1986)
8.1.1.1	Carlow (1992); Smith and Mosier (1986)
8.1.1.2	Carlow (1992); Smith and Mosier (1986)
8.1.1.3	Smith and Mosier (1986)
8.1.1.3.1	Carlow (1992)
8.1.1.4	Carlow (1992)
8.1.1.5	Carlow (1992); Smith and Mosier (1986)
8.1.2	Carlow (1992); Smith and Mosier (1986)
8.1.3	Smith and Mosier (1986)
8.1.4	Smith and Mosier (1986)
8.1.4.1	Smith and Mosier (1986)
8.1.4.2	Smith and Mosier (1986)
8.2	Carlow (1992); Smith and Mosier (1986)
8.2.1	Carlow (1992); Smith and Mosier (1986)
8.2.2	Carlow (1992); Smith and Mosier (1986)
8.2.3	Carlow (1992); Smith and Mosier (1986)
8.3	Carlow (1992); Smith and Mosier (1986)
8.3.1	Carlow (1992)
8.3.2	Carlow (1992)
8.3.2.1	Carlow (1992); Shneiderman (1992)
8.3.2.2	Carlow (1992); Smith and Mosier (1986)
8.3.3	Carlow (1992)
8.3.4	Carlow (1992); Smith and Mosier (1986)
8.3.5	Smith and Mosier (1986)
8.3.5.1	Smith and Mosier (1986)
8.4	Carlow (1992)
8.4.1	Carlow (1992); Smith and Mosier (1986)
8.4.1.1	Carlow (1992)
8.4.2	Carlow (1992)
8.4.3	Carlow (1992); Smith and Mosier (1986)
8.5	Smith and Mosier (1986)
8.5.1	Smith and Mosier (1986)
8.5.2	Smith and Mosier (1986)
8.5.3	Horton (1990); Shneiderman (1992); Galitz (1993)
8.5.3.1	Horton (1990); Shneiderman (1992)
8.5.3.2	Horton (1990); Galitz (1993)
8.5.3.3	Horton (1990)
8.5.3.4	Horton (1990)
8.5.3.5	Horton (1990)

GLOSSARY

Accelerators keyboard commands that can be used instead of pointing and clicking on menu options.

Active voice a mode of writing in which sentences are constructed so that a person or thing (the subject) acts directly rather than indirectly, usually upon an object; for example, the sentence, "The user opened a file," is in the active voice, but "A file was opened by the user," is in the passive voice.

Application area screen space that serves as the user's main work space; also called the screen body or client area.

Attribute a user-specified characteristic, such as color, line length, or size (Smith & Mosier, 1986).

Audible signal an alarm, alert, caution, or warning that is presented to the user's auditory channel; for example, a beeping sound that signals an invalid data entry.

Automatic panning an approach to managing multiple windows in which panning is specified from bottom to top; for example, a new window is presented at the bottom of the screen, forcing upwards the window previously displayed at the bottom, which pushes the topmost window off the display (Shneiderman, 1992).

Brightness the perceptual experience of the intensity of light, varying from bright to dark (Thorell & Smith, 1990); indistinguishable from lightness on a color CRT (Mayhew, 1992).

Cascade buttons pushbuttons that open pull-down menus when activated: the label on the cascade button serves as the title for the pull-down menu.

Cascades an application of the deck-of-cards metaphor to management of a sequence of windows; typically, each successive window is offset below and to the right of the previous window, leaving the window titles visible (Shneiderman, 1992).

Cases a letter having as its typical form either the lower-case, e.g., *a f g* or *b n i* , or the upper-case, e.g., A F G or B N I.

Check box a type of pushbutton implemented in software; behaves as an on-off toggle to indicate that one or more options is on or off; presenting a group or panel of check boxes allows the user to make multiple choices; also called check button; contrast with radio button.

Coding 1) use of visual "codes," such as size, shape, or color to highlight or emphasize displayed data or objects; 2) use of a programming language to build software.

Cognitive engineering the process of designing a user interface for compatibility with human capabilities and with the user's tasks (Rasmussen, 1986; Woods & Roth, 1988).

Cognitive task analysis the process of identifying and describing work activities that involve primarily mental effort, such as problem solving and decision making; the methods used to perform such an analysis (Rasmussen, 1986; Schlager, Means, & Roth, 1990).

Color blindness see Color deficiency.

Color contrast the ratio of the color (hue) and saturation of a foreground image compared to the color (hue) and saturation of its background (Thorell & Smith, 1990); adequate color contrast is necessary for legibility; adequacy depends on lighting conditions.

Color deficiency a reduced ability to detect and distinguish some colors; 8-10% of Caucasian males, 4% of non-Caucasian males, and .5% of all females are born with some variation of the possible deficiencies in color vision (Thorell & Smith, 1990).

Color discrimination the user's ability to distinguish colors from each other; varies with color contrast and lighting conditions.

Command area a space into which the user can type an instruction (command) to the computer.

Consistency the level of predictability in the layout and behavior of a user interface; a key principle in user-interface design.

Context-sensitive help detailed information linked to the cursor's current location in an error field, provided upon request from the user (Mayhew, 1992).

Control entries commands, function keys and icons that are selected for a particular transaction.

Current focus the location on the screen to which the user is directing attention; usually the active window.

Data validation an internal process whereby data entered by the user are examined automatically for such attributes as correct format and acceptable values; produces an error message to the user when data are found to be invalid.

Data-field label a displayed word or phrase that serves as a prompt for entering an item in a data field; such a label usually cannot be changed by a user (Smith and Mosier, 1986).

Dialog any exchange of data/information between the user and the computer, from log-on to log-off; an interaction between the user and the computer.

Dialog boxes windows used to ask the user to complete an action within a limited spatial context (Galitz, 1993).

Direct access a method of *rapid access*, which allows the experienced user to access terminal options in a menu structure by entering a mnemonic label for the target, bottom-level menu option (e.g., DDF for Data Distribution Facility); an "all-or-nothing" *jump-ahead* strategy (Norman, 1991, p. 254).

Direct manipulation a form of rapid, incremental, reversible interaction that gives the user a powerful sense of acting on the displayed environment (Shneiderman, 1992).

Display control procedures that allow the user to specify what data are to be displayed and how the data are to be displayed (Smith & Mosier, 1986).

Emphasis techniques methods of displaying data so that the data's importance or criticality is stressed; used to attract the user's attention to specific areas of the visual display; includes highlighting, blinking, and color coding.

Error tolerance 1) an automated capability for capturing input errors that pass through the data-validation process before they propagate throughout the system; 2) acceptance of close approximations or synonyms for standard commands.

Feedback 1) guidance provided in response to user action for the purpose of helping the user navigate through the display system and understand error messages; may take the form of a warning, a dialog with the user to clarify and resolve a problem, or a query to probe user intentions (Lewis & Norman, 1986; Mayhew, 1992); 2) information about system status; 3) a visual echo of each keystroke.

Focus the portion of the screen that displays current user input; another term for the active window.

Font typeface or typestyle; for example, Courier, Times, Helvetica; characterized by the presence or absence of serifs.

Form-fillin displays a series of captioned blank fields to be filled with required information; the user is expected to understand the captions, know the method of entry and the permissible values to be keyed, and be able to respond to and correct errors.

Formatting the organized presentation of display elements, including labels and user guidance; see also Grouping.

Framing 1) presentation of task-related data in one application area or page, to minimize the need to scroll or page through a set of displays; requires that data be grouped on the basis of task-related information needs; 2) user display of data coverage by display movement, including paging, scrolling, offset, expansion, etc. (Smith and Mosier, 1984).

Graphical user interface (GUI) as compared to a textual user interface, uses geometrical shapes, symbols, icons, and lines of various types to convey information.

Grouping screen layout based on the clustering of related data; individual clusters are separated from others by white space or by lines drawn around groups; if task-appropriate, promotes user efficiency.

Hierarchical menus structures of menu options that branch downwards from one top-level (root) option; several different pathways are possible; selection of an option determines which options will follow in sequence (Mayhew, 1992; Norman, 1991).

Highlighting calling attention to a portion of displayed data or format features by underling, bolding, shadowing, boxing, reverse video, blinking, or applying color; specifically, an emphasis technique typically used to distinguish the current selection in a menu from the other menu options; usually implemented by increasing the intensity (brightness) of the highlighted item or by displaying it in reverse video.

Icon graphical objects that can be selected, can represent commands, or can represent applications or tools; selecting an icon, issues a command to the application software or, if the application is iconified, opens the application.

Incremental positioning cursor movement that is horizontally and vertically consistent in step size (Smith and Mosier, 1986).

Intensity with regard to light, the number of light particles (photons) emitted by a source of light (Thorell & Smith, 1990).

Interaction in the context of a user interface, any exchange of data/information between the user and the computer system; for example, the user keys in text, which is echoed on the video display; the user selects a menu option, and that option is activated; supported by interaction devices, such as keyboard, mouse, or trackball; see also Transaction.

Jump ahead a technique to allow expert users to by-pass middle levels of a menu structure; may be implemented by *direct-access* or *type-ahead* strategies (Norman, 1991).

Justification in lists or columns, the issue of how to align items under each other and under a heading; for example, a left-justified list is lined up at the left-most edge (margin, border) of the display space; alternative choices are right-justified or centered, although left justified is generally preferred.

Labeling providing a title or descriptor that helps a user identify displayed data.

Landmarks in the context of a menuing system, nodes that are frequently accessed and that represent transition points between states or kinds of environments; like geographical landmarks or anchor points that the user can return to when restarting a search; also called "cognitive landmarks" (Norman, 1991, p. 230).

Leading zeros zeros that must precede numbers because of programming or processing conventions; if leading zeros are required, the user must enter, for example, 0005 instead of 5; contrasted with trailing zeros, for example, 5.00 instead of 5 with no decimal point or following zero place holders.

Lightness the perceptual experience of the achromatic content of a visual image, varying from light through gray to black (Thorell & Smith, 1990).

Long-term memory the "data base" of knowledge in the brain; a critical component of human information processing; its capacity for storing knowledge is assumed to be infinite; contrast with short-term memory.

Lower-case a non-capitalized letter having as its typical form*a f g* or *b n i* .

Main window see primary window.

Menu a set of related options listed together for selection by the user; a typical feature of a graphical user interface; see also pop-up, pull-down, and tear-off menus.

Menubar a panel, usually located at the top of the screen, that continually displays the top-level menu options for selection.

Message area a portion of screen space, preferably located at the bottom or center of the screen, in which status information is displayed; can be displayed only when there is a message to present.

Message box a type of dialog box that displays a message in either a message line or pop-up window.

Mnemonic code short command name or code that is easy to learn and remember.

Mode a particular functioning arrangement, state or condition.

Navigation in the context of menuing systems, the user's ease of moving forward and backward through the menu structure in pursuit of target menu options. In general, the user’s ability to move easily and maintain a sense of orietation within the user interface structure (e.g. from window to window).

Non-space-filling tiling a tiling strategy that does not require covering the whole screen with tiled windows; gaps are allowed between tiles; overlaps are not allowed; overcomes the problem of narrow windows (Shneiderman, 1992).

Option-Button menu a list of related options, implemented in a single line with a scrolling button or arrow to the right of the topmost options; when the user clicks on the scrolling button or arrow, the full list of options opens beneath the topmost option.

Output display output of data from a computer to its user.

Overlapping windows windows are like papers on a desktop, arranged in a simulated three-dimensional space and, perhaps, stacked in piles, overlapping on another (Mayhew, 1992).

Paging sequential movement of displayed data by full screen units.

Palette a set of unlabeled symbols, typically presented within small rectangles; a maximum of six or seven foreground and background colors for color variety so that the user will have the opportunity to change colors if it is desired.

Panning a set of system-controlled, consistent strategies for moving open windows and placing new windows on the screen; in map displays, the process of smoothly and predictably changing the displayed region.

Passive voice as compared to active voice, a less direct and often more wordy way of describing the activity of some agent; for example, "Help was requested by the user," is a passive versions of "The user requested help." Sometimes necessary, for example, if the performer(s) of some action wish to remain anonymous or cannot be identified: "A decision was made to eliminate 20 positions in the division"; "The bank was robbed at gunpoint."

Pie menus a pop-up menu design technique to minimize the distance from the cursor to the target option and to maximize the size of the target option (Callahan, Hopkins, Weiser, & Shneiderman, 1988; Norman, 1991; Shneiderman, 1992).

Piles of tiles a tiling strategy in which windows completely overlap because they are stacked on top of one another; tiles are "popped" to access windows lower in the stack; several piles of tiles can be in use simultaneously, with a different number of tiles in each pile (Shneiderman, 1992).

Pop-up a list of options or menu that is displayed when the user selects a specially-programmed graphical symbol or "hot spot" on the computer screen; serves to save screen space since it is not displayed continuously.

Primary window the highest-level window in an application; displays the essential data that the user requires to interact with the application; also called the main window.

Proximity a visual coding technique in which closeness of placement indicates degree of relatedness.

Pull-down a list of sub-menu options that is displayed when the user selects a higher-level menu title, such as File, Edit, or View; displays the options available under the selected menu title; also called a drop-down menu.

Pushbuttons in the context of the user interface, software-generated objects that are displayed on the computer screen; often designed to resemble physical pushbuttons through outlining and shadowing; activated by pointing and clicking, or in a touchscreen application, by physically touching the appropriate screen space; also called command buttons.

Radio button a type of electronically-displayed pushbutton used for exclusive selection; the user can select one, and only one, option from a set of radio buttons; contrast with check box.

Rapid access a capability provided for the experienced user to bypass intermediate levels of a menu structure (Norman, 1991).

Recall unaided retrieval of facts or other information from long-term memory; typically more difficult than recognition because no cues are given.

Recognition following presentation of a cue, such as a picture, the matching of the cued information to information in long-term memory.

Reverse video the inversion of normal display brightness; a highlighting technique.

Saturation the percentage of pure color content (Thorell & Smith, 1990); for example, dark blue is highly saturated, but light blue is low in saturation (i.e., a little blue is mixed with white).

Scrolling an orientation for display framing, such that the data appears to be moving behind a fixed display frame.

Secondary window a transient or child window associated with a primary window.

Selection field a field that displays on the screen all the possible alternatives, conditions or choices the may exist for a data element or action (Galitz, 1993).

Sequence control the set of user actions and related computer logic that govern the transitions between user-computer transactions (Smith & Mosier, 1986).

Short-term memory (STM) a limited-capacity system that holds and manipulates the currently active contents of consciousness.

Special characters non-alphanumeric characters such as the at sign (@) and backslash(\).

System map a small, schematic overview of an application's display system; used to aid the user in navigating through the display system.

Tabular data numerical values presented under column headings or in a matrix format of labeled rows and columns.

Task a unit of mental or physical work that has a definable start and finish, that is made up of several smaller units (sub-tasks or task elements), and that consumes time on the order of minutes (not seconds or hours); a sub-unit of a function; can be characterized in terms of information requirements and the knowledges, skills, abilities, and other attributes needed by the worker; may be associated with management expectations for speed and accuracy (performance requirements).

Task analysis the process of identifying the component parts of a job, that is, the work activities required to fulfill the responsibilities of a position or set of positions (usually in terms of a hierarchy of functions, tasks, and sub-tasks); typically includes identification of the information required to perform work activities as well as the knowledges, skills, abilities, and other worker attributes needed for acceptable performance; methods are tailored to the particular purposes and circumstances of the analysis.

Tear-off menu a menu that can be "torn" from the menubar and moved to another location on the screen.

Text field a specified area where alphanumerics, written words, sentences, and/or paragraphs are entered, edited, and displayed.

Tiled windows defined display areas that do not overlap with each other.

Transaction a user action that is followed by a computer response; equivalent to Interaction (Smith & Mosier, 1986).

Transaction sequence the order of user and computer actions that make up a defined exchange of information.

Transparency the quality of being self-explanatory, predictable, or intuitive; when an interface is transparent to the user, the user is able to concentrate on the task at hand does not have to be concerned with software structures underlying the interface; a key principle of user-interface design.

Type ahead a method of *rapid access*, which allows an expert user to bypass middle levels of a menu structure by entering several sequential levels at once; provides partial *jump ahead* if the user can recall only part of the required sequence (Norman, 1991).

Update recomputation and display of dynamically changing data to inform the user of current status; may occur by user request or be performed automatically (Smith & Mosier, 1986).

Upper-case capital letters, such as A F G or B N I.

Usability the quality of being not only functional but also suitable to the user's task and easy to learn; a key principle of user-interface design.

Usability testing a systematic approach to evaluating a user interface for its effects on user productivity and satisfaction.

User guidance computer-generated error messages, alarms, prompts, and instructional materials designed to aid the user in navigating through the user interface; see also Feedback.

User interface the set of displays and controls that support interaction (information transfer) between the user and the computer system; permits the display of information to the user and the user's input of data needed by the computer; also called the computer-human interface (CHI), the human-computer interface (HCI), or the user-system interface (USI); formerly known as the man-machine interface (MMI).

User productivity the level of the user's performance in terms of measures such as speed and accuracy.

User-centered perspective an approach to user-interface design that seeks to understand the user's goals, functional requirements, and tasks before building the interface; as contrasted with a technology-centered perspective, which focuses primarily on hardware and coding considerations.

User-developed macro a small file created by the user, which contains the sequence of keystrokes required to navigate through the menu structure to a specific destination; a method to facilitate an expert user's navigation of a menu structure (Mayhew, 1992).

Window a defined area of display space; a data structure that is usually represented as a rectangular portion of the screen.

Window controls soft buttons that allow the user to change window attributes.

Window frame the bordered area around a window.

Window zooming a feature that allows the user to expand and shrink windows; allows many tiled windows to be displayed simultaneously in their shrunken or "iconified" state (Shneiderman, 1992).

REFERENCES

American National Standards Institute (ANSI)/Human Factors Society (HFS). (1988). *American national standard for human factors engineering of visual display terminal workstations* (ANSI/HFS Standard No. 100-1988). Santa Monica, CA: Human Factors Society.

Apple Computer. (1987). *Human interface guidelines: The Apple desktop interface*. New York: Addison-Wesley.

Avery, L. W., & Bowser, S. E. (Eds.). (1992). *Department of Defense human-computer interface style guide* (Version 2.0; DTIC # A252410). Fort Lewis, Washington: U. S. Army.

Bailey, R. W. (1982). *Human performance engineering: A guide for system designers*. Englewood Cliffs, NJ: Prentice-Hall.

Billingsley, P. A. (1988). Taking panes: Issues in the design of windowing systems. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 413-436). Amsterdam: Elsevier/North Holland.

Bleser, T. W. (1994). *Human-computer interface guidelines evaluation*. Laurel, MD: Century Computing.

Brown, C. M. (1988). *Human-computer interface design guidelines*. Norwood, NJ: Ablex.

Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988). An empirical comparison of pie vs. linear menus. *CHI'88 Conference Proceedings on Human Factors in Computing Systems* (pp. 95-100). New York: Association for Computing Machinery.

Campbell, R. (1994). *Client-server systems user interface style guide* (DSTL-95-DRAFT). Greenbelt, MD: NASA/Goddard Space Flight Center.

Carlow International Incorporated. (1992). *Human-computer interface guidelines* (DSTL-92-007). Greenbelt, MD: NASA/Goddard Space Flight Center.

Carter, R. C., & Carter, E. C. (1982). High-contrast sets of colors. *Applied Optics*, *21*(16), 2936-2939.

Chignell, M. H., Hancock, P. A., & Lowenthal, A. (1989). An introduction to intelligent interfaces. In P. A. Hancock and M. H. Chignell (Eds.), *Intelligent interfaces: Theory, research and design* (pp. 1-26). New York: Elsevier/North Holland.

Christ, R. E. (1975). Review and analysis of color coding research for visual displays. *Human Factors*, *17*(6), 542-570.

Department of Defense (DoD). (1989). *Human engineering design criteria for military systems, equipment and facilities* (MIL-STD-1472D, Notice 1). Washington, DC: DoD.

Dondis, D. (1973). *A primer of visual literacy.* Cambridge, MA: MIT Press.

Dumas, J. (1988). *Designing user interfaces for software.* Englewood Cliffs, NJ: Prentice Hall.

Fowler, S. L., & Stanwick, V. R. (1995). *The GUI style guide.* New York: Academic Press.

Galitz, W. O. (1993). *User-interface screen design.* New York: John Wiley.

Gilmore, W. E., Gertman, D. I., & Blackman, H. S. (1989). *The user-computer interface in process control: A human factors engineering handbook.* New York: Academic Press.

Gould, J., & Grischkowsky, N. (1984). Doing the same work with hardcopy and with cathode ray tube (CRT) terminals. *Human Factors*, *26*, 323-337.

Grudin, J. (1989). The case against user interface consistency. *Communications of the ACM, 32,* 1164-1173.

Hix, D., & H. R. Hartson. (1993). *Developing user interfaces: Ensuring usability through product & process.* New York: John Wiley.

Horton, W. (1991). *Illustrating computer documentation: The art of presenting information graphically on paper and online.* New York: John Wiley.

Horton, W. K. (1990). *Designing and writing online documentation.* New York: John Wiley.

Horton, W. K. (1994). *The icon book.* New York: John Wiley.

International Organization for Standardization. (1993). *Ergonomic requirements for office work with visual display terminals* (Part 8: Requirements for displayed colours, Draft ISO/DIS 9241-8; Part 10: Dialogue principles, Draft ISO/DIS 92412-10; Part 11: Guidance on specifying and measuring usability, Draft ISO/CD 9241-11.3; Part 13: User guidance, Draft ISO/CD 9241-13). Hackensack, NJ: American National Standards Insitute.

International Organization for Standardization. (1994). *Ergonomic requirements for office work with visual display terminals* (Part 12: Presentation of information, Draft ISO/CD 9241-12). Hackensack, NJ: American National Standards Institute.

Kobara, S. (1991). *Visual design with OSF/Motif.* New York: Addison-Wesley.

Lanier, J. (1995). Agents of alienation. *interactions, II.3,* 66-72.

Lewis, C., & Norman, D. A. (1986). Designing for error. In D. A. Norman & S. W. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction* (pp. 411-432). Hillsdale, NJ: Erlbaum.

Marcus, A. (1992). *Graphic design for electronic documents and user interfaces.* New York: ACM Press.

Mayhew, D. J. (1992). *Principles and guidelines in software user interface design.* Englewood Cliffs, NJ: Prentice Hall.

Meister, D. (1985). *Behavioral analysis and measurement methods.* New York: John Wiley.

Mitchell, C. M., & Saisi, D.L. (1987). Use of model-based qualitative icons and adaptive windows in workstations for supervisory control systems. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-17*(4), 573-593.

Mullet, K. (1995). Organizing information spatially. *interactions, II.3,* 15-19.

Murch, G. M. (1987). Color displays and color science. In H. J. Durrett (Ed.). *Color and the computer* (pp. 1-25). New York: Academic Press.

Norman, D. A., & Draper, S. W. (Eds.) (1986). *User centered system design.* Hillsdale, NJ: Erlbaum.

Norman, K. L. (1991). *The psychology of menu selection: Designing cognitive control at the human/computer interface*. Norwood, NJ: Ablex.

Norman, K. L., Weldon, L. J., & Shneiderman, B. (1986). Cognitive layouts of windows and multiple screens for user interfaces. *International Journal of Man-Machine Studies*, *25*, 229-248.

Open Software Foundation. (1991). *OSF/Motif style guide* (Revision 1.1). Englewood Cliffs, NJ: Prentice Hall.

Paap, K. R., & Roske-Hofstrand, R. J. (1988). Design of menus. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 205-235). Amsterdam: Elsevier/North Holland.

Paap, K. R., & Roske-Hofstrand, R. J. (1986). The optimal number of menu options per panel. *Human Factors*, *28*, 377-385.

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. New York: Elsevier/North Holland.

Schlager, M. S., Means, B., & Roth, C. (1990). Cognitive task analysis for the real(-time) world. *Proceedings of the Human Factors Society 34th Annual Meeting* (pp. 1309-1313). Santa Monica, CA: Human Factors Society.

Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction* (2nd ed.). New York: Addison-Wesley.

Siebert, L., & Ballard, L. (1992). *Making a good layout.* Cincinnati, OH: F&W/North Light Books.

Silverstein, L. D. (1987). Human factors for color display systems: Concepts, methods, and research. In H. J. Durrett (Ed.), *Color and the computer* (pp. 38-51). New York: Academic Press.

Smith, S. L., & Mosier, J. N. (1986). *Guidelines for designing user interface software.*(ESD-TR-86-278) Bedford, MA: MITRE.

Thorell, L. G., & Smith, W. J. (1990). *Using computer color effectively.* Englewood Cliffs, NJ: Prentice Hall.

Thurman, D. A. (1995). *Improving operator effectiveness in monitoring complex systems: A methodology for the design of interactive monitoring and control interfaces* (Unpublished Master's thesis). Atlanta, GA: Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology.

Thurman, D. A., & Mitchell, C. M. (1995). A design methodology for operator displays of highly automated supervisory control systems. *Preprints of the 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems* (pp. 821-826). Cambridge, MA: IFAC/Massachusetts Institute of Technology.

Travis, D. (1991). *Effective color displays: Theory and practice.* New York: Academic Press.

Tufte, E. R. (1990). *Envisioning information.* Cheshire, CT: Graphics Press.

Tullis, T. S. (1981). An evaluation of alphanumeric, graphic and color information displays. *Human Factors*, *23*, 541-550.

Tullis, T. S. (1988). Screen design. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 377-411). Amsterdam: Elsevier/North Holland.

Uehling, D. (1994). *Usability testing handbook* (DSTL-94-002). Greenbelt, MD: NASA/Goddard Space Flight Center.

Whiteside, J., Bennett, J., & Holtzblatt, K. (1988). Usability engineering: Our experience and evolution. In M. Helander (Ed.) *Handbook of human-computer interaction* (pp. 791-817). New York: Elsevier/North Holland.

Wickens, C. D. (1992). *Engineering psychology and human performance* (2nd ed.). New York: HarperCollins.

Woods, D. D. (1984). Visual momentum: A concept to improve the cognitive coupling of person and computer. *International Journal of Man-Machine Studies, 21*, 229-244.

Woods, D. D., & Roth, E. M. (1988). Cognitive systems engineering. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 3-43). New York: Elsevier/North Holland.

Wyszecki, G., & Stiles, W. S. (1982). *Color science: Concepts and methods, qualitative data and formulae* (2nd ed.). New York: John Wiley.

Appendix A

User-Interface Design and Evaluation Methods

This appendix presents a brief overview of the user-interface design process as practiced in Code 520. The discussion focuses on the design of the user interface as separate from development of the underlying functionality. Although the process is presented as a series of steps, some steps may be pursued in parallel, and each step may be revisited several times during user-interface development.

Step 1. Define Requirements

Solicit initial user-interface requirements from the sponsoring organization. Requirements should be device independent and testable. If end users have not contributed to these requirements, work with the sponsoring organization to identify a group of end users to review the requirements.

Step 2. Form a Users' Group

Purpose. The function of the users' group is to provide an operational perspective on both the requirements and proposed displays and controls.

Membership. Members of the users' group include potential end users of the system, who may be end users of the current system (if there is one) or end users of a similar existing system. Other members of the user's group include members of the development team and key decision makers from the sponsoring organization. One or more experts in Human-Computer Interface (HCI) design may participate in meetings of the users' group. Each member should have some understanding of end users' operational requirements and procedures, including approaches to problem solving and decision making.

Agenda and Minutes. Meetings of the users' group are held frequently to review and interpret stated requirements and to derive further requirements. Once past the requirements-definition stage, the users' group continues to meet to review displays and controls proposed to implement the requirements. Having an agenda for each meeting helps to structure the discussion and provide a record. Compiling minutes of the meetings provides documentation of decisions and the rationale behind them. Action items and the responsible parties can be included in the minutes.

Step 3. Profile End User Types

The development team leads the effort to define categories of end users and their roles in the system. The profile of each end-user type includes the category name (e.g., data base administrator, scheduler), the role played in the system, and the set of system privileges assigned to that category.

Step 4. Derive Operational Scenarios

Scenarios are needed to support development of an operations concept and to permit systematic usability testing. Scenarios comprise sequences of operational activities for each end-user type. The activities included in a scenario represent frequently performed tasks that require the user to exercise a broad range of system and user-interface capabilities. Examples of scenarios are presented in Exhibit A-1.

Draft scenarios are reviewed in detail during meetings of the users' group. Iterative review and revision of the scenarios continues until the users' group reaches consensus on the system model embodied in the user interface. The scenarios are included in an operations concept document.

Step 5. Identify Major Components

Categories of information and interaction are the major components of the user interface. Information comprises the data elements and groupings of data to be displayed to the user. Interactions comprise the user actions to be supported by the application (e.g., search, monitor, select, plan). Data elements may be alphanumeric and graphical. Interactions may be highly cognitive, that is, involving problem solving and decision making, as well behavioral.

Identification of major components is supported by task analysis, which focuses on defining the functions, tasks, and task actions to be performed by the user. It is important to define both cognitive and behavioral tasks. The user's information requirements are typically defined at the task level. An information-flow analysis may be conducted to capture the exchange of information between user tasks.

Step 6. Prototype the User Interface

The first prototyping activity is the creation of a paper prototype. It is helpful to have two developers work separately to generate paper mockups of major system displays based on the requirements. The development team and HCI expert(s) then meet to consider the paper prototypes and pick the best features of each design.

Next, an initial prototype of the user interface is created online by implementing the selected design features in accordance with usability guidelines. Simple heuristic evaluations may be conducted by user-interface experts. To obtain additional feedback, prototypes are demonstrated to members of the User's Group. Demonstrations follow representative scenarios (from Step 4). Alternatively, packets of screen printouts may be reviewed in full meetings of the User's Group.

In the stand-alone system environment, detailed specifications for UI components are developed prior to prototyping. Exhibit A-2 presents an example of a window specification. The prototyping and review process may be more or less formal, depending on the management issues defined by the size and scope of the system under development.

Through cycles of prototyping, review, and modification, the user-interface prototype becomes increasingly detailed. This process continues until developers and users feel comfortable with the design of the displayed information and the dynamics of user-computer interactions. The final prototype becomes an integral part of the final system.

Step 7. Develop a Project Style Guide

A style guide is needed to encourage consistency across different developers who may be working on different UI components. The style guide specifies the standard design for UI components, as defined by developers and HCI experts.

As described in Section 1 of this document, a project style guide is based on a selected subset of the higher-level UI guidelines. Creation of the style guide begins during prototyping and may continue through development as issues arise. The style guide may be thought of as a "living document," subject to updates that resolve conflicting guidelines or other issues.

The process of developing a style guide is outlined in the introduction (Section 1). A group composed of the project lead, the HCI experts, and a documentation expert serves as the forum for review and revision of the style guide.

The style guide is followed religiously by developers during the coding process.

Step 8. Test for Usability

Usability testing is essential to discover any problems that may have been overlooked during design and development. The goal of usability testing is to determine whether an application's user interface makes it relatively easy and natural for the user to perform required tasks. Code 522's Usability Testing Handbook (Uehling, 1994) provides detailed guidance on how to conduct usability tests. Developers fix any problems in usability or functionality identified through usability testing. Then the system is retested, modified if necessary, and released.

Exhibit A-1
ASAP Operational Scenarios

1 User Logs into ASAP

1 User selects icon for ASAP from Windows

2 ASAP displays dialog box for entry of userid and password

3 User enters valid userid and password and indicates OK

4 ASAP displays first application window containing user's inbox

2 Log into ASAP Falls

1 User selects icon for ASAP from Windows

2 ASAP displays dialog box for entry of userid and password

3 User enters invalid userid or password and indicates OK

4 ASAP generates message that userid and/or password is invalid

5 ASAP redisplays log in dialog box for up to three attempts at logging in

6 After three failed attempts, ASAP generates message that log in has failed and that the user must contact the ASAP system administrator

3 Browse User's Inbox PR's

1 User logs into ASAP (scenario 1)

2 ASAP displays user's inbox
  2.1 List will be sorted by ...(TBD)
  2.2 List will contain PR's based on the following:
    3.2.2.1 If the user has initiated PR's
      (PR's which have been initiated by the user, but not yet released
      (PR's which have been returned to the initiator by resources analyst or approval official
    2.2.2 If the user is a resources analyst (RA)
      PR's which have been released to the RA
      PR's which have been returned to the RA by procurement
    3.2.2.3 If the user is an approval official

PR’s needing approval/signature by this official

3 User may scroll through the list to view all PR’s in inbox

4 ASAP will indicate the current status of each PR in the list

5 User may select a PR to display summary information, currently first item (TBD)

6 User may indicate an action on a selected PR (e.g., view, edit, approve, release, see history, etc.). Only those actions which are valid for the status of the PR will be available. Action may not be indicated for a selected PR if the PR is already open in detail window

4 Browse Other PR’s Accessible by User (i.e., other than inbox)

1 User logs into ASAP (scenario 1)

2 ASAP defaults to displays list of user’s inbox (scenario 3)

3 User selects type of list to view

3.1 User may select Outstanding PR’s - not obligated (PO not awarded)
3.1.1 If the user has initiated PR’s
(PR’s which have been initiated by the user, not yet obligated (PO not awarded)
3.1.2 If the user is an (RA)
PR’s which have been released to the RA
PR’s which have been returned to the RA by procurement
4.3.1.3 If the user is an approval official
PR’s needing approval/signature by this official

3.2 User any selection retired PR’s - fully received and paid, associated PO has been retired by buyer
3.2.1 If the user has initiated PR’s
PR’s which have been initiated by the user
4.3.2.2 If the user is a resources analyst (RA)
PR’s which have been assigned this RA
4.3.2.3 If the user is an approval official
PR’s which this official approved

3.3 User may select al PR’s

3.3.1 If the user has initiated PR’s
PR’s which have been initiated by the user
4.3.3.2 If the user is a resources analyst (RA)

PR's which have been assigned this RA

4.3.3.3 If the user is an approval official
PR's which this official approved

4 User may select a PR to display summary information, currently first item (TBD)

5 User may select list sorting parameters (TBD)

6 User may indicate an action on a selected PR. Allowable actions are dependent on the status of the PR. Action may not be indicated for a selected PR if the PR is already open in detail window.

5 Enter a New Purchase Request - valid information

1 User logs into ASAP (scenario 1)

2 ASAP displays list of user's inbox (scenario 3)

3 User indicated New PR option

4 ASAP opens window for PR detail window with header reading PCN UNASSIGNED

5 User enters PR detail information (scenario 3)

6 User indicates saves PR

7 ASAP saves PR

7.1 Determines next available PCN-suffix, concatenates with ORG for PCN

7.2 ASAP checks AMS for potential PCN, AMS does not already contain the specified PCN

7.3 ASAP assigns PCN to PR

7.4 Record is stored in database, ASAP logs date and time of record creation

8 ASAP displays new PCN on title of window

6 User cancels new PR

1 Same as scenario 5.1 - 5.4

2 User indicates Close, without having saved PR

3 ASAP returns to the PR list window and the PR is not saved

7 User cancels new PR - PCN suffix range used up

1 Same as scenario 5.1 - 5.4

2 User indicates Save PR

3 ASAP displays a message that the PCN is out of range, too contact the system administrator

4 ASAP will store the PR with a temporary PCN for later saving and assignment of the permanent PCN

8 User Enters PR Detail Information

1 ASAP displays window containing access to all PR information, beginning with Header information

2 The user may select specific categories of PR information to enter in any sequence

2.1 Header - general information pertaining to the purchase request (scenario 9)
2.2 Delivery - information pertaining to the individual and address where the goods will be shipped (scenario 10)
2.3 Items - the specific items to be purchased (scenario 11)
2.4 Sources - suggested sources for the procurement of the goods (scenario 12)
2.5 Text - attachments to the PR (scenario 13)
2.6 JON - accounting information (scenario 14)
2.7 Special - special approvals and hazard information (scenario 15)

3 User may provide Notes attached to the PR (TBD)

9 User Enters PR Header Information

1 ASAP provides default values

1.1 Name, ORG code and phone number of initiator and technical representative are pre-filled identifying individual being acted for (required)

1.2 Accept days is defaulted to 7 (required)

1.3 Location indicator is defaulted too Greenbelt or Wallops, depending on the location of the individual being acted for (required)

2 User provides values for non-default fields, and may change default values

2.1 Resources Analyst name, ORG and phone (a list box will be provided for selection of values) (required)

2.2 Date by which the goods are needed (optional)

2.3 Keyword (a list box specific to each location will be provided for selection of values) (required)

2.4 Total PR amount - this amount will be computed based o the item total for SEWP PR's, amount must be entered for non-SEWP PR's (required)

2.5 The 'S number' if this PR amends a purchase order (optional)

2.6 Flags - SEWP, Imprest, Emergency and Spaceflight (required)

2.7 (If SEWP - do we want to show SEWP contract number?)

10 User Enters PR Delivery Information

1 User enters name, ORG, phone number of individual to whom the goods are to be shipped (default value TBD)

2 ASAP pre-fills the full receiving address to be the address of the receiving dock for the user's location, Greenbelt or Wallops - the user may modify if necessary or indicate selection to reload from default address

11 User Enter PR Item Information - non SEWP

1 ASAP requests specific information identifying each item to be purchased

2 ASAP requests specific information identifying each item to be purchased

3 The user may enter item information from 'scratch'

3.1 The user must enter specific required information

11.3.1.1 Quantity - defaulted to 1

11.3.1.2 Unit - select from pick list

11.3.1.3 Description of the Item

11.3.1.4 Type of Procurement - select from pick list

11.3.1.5 Keyword - select from pick list

Exhibit A-2
SPSR Window Specifications

Name
SLR Overview (SLR id)

Purpose
This window provides the operator with an overall summary of each category of resources and indicators conveying the current status of those resources. These indicators will clearly show the operator if the resources in that category are unavailable, changed, or have remained the same. Another indicator will clearly show whether a SHOs List or if a SHO has been removed from the list. Figure A-2 illustrates the NCC DS Service Planning Segment Replacement.

Type
Standard

Display
Date SLR sent
SLR id - title of the window
Ground Terminal that sent the SLR

List containing each resource category

- SGLT 1 Service Chains
- SGLT 1 Computer Subsystems
- SGLT 1 TT&C Chains
- SGLT 2 Service Chains
- SGLT 2 Computer Subsystems
- SGLT 2 TT&C Chains
- SGLT 3 Service Chains
- SGLT 3 Computer Subsystems
- SGLT 3 TT&C Chains
- Antennas
- DIS Subsystem
- TDRS

Beside each will be a checkbox indicating if there has been a change since the last SLR was received.

Each category will also have an indicator showing what type of change in status has occurred. A red indicator (icon?) would indicate a resource in that category is now unavailable. A yellow indicator would convey that a backup of that event has went down.

Another indicator will show if a SHO has been added to the Affected SHO List or if a SHO has been removed from the list.

Control

The operator will be able to double click on a resource catagory to get a more detailed look at which resources have changed.

Operation
TDRSS Status (SRD 5.3.7)

Opened by
Alert Window

Opens
Service Chains for SGLT 1
Service Chains for SGLT 2
Service Chains for SGLT 3
Computer Subsystems for SGLT 1
Computer Subsystems for SGLT 2
Computer Subsystems for SGLT 3
TT&C Chains for SGLT 1
TT&C Chains for SGLT 2
TT&C Chains for SGLT 3
DIS Subsystem
Antenna
TDRS
Affected SHO List

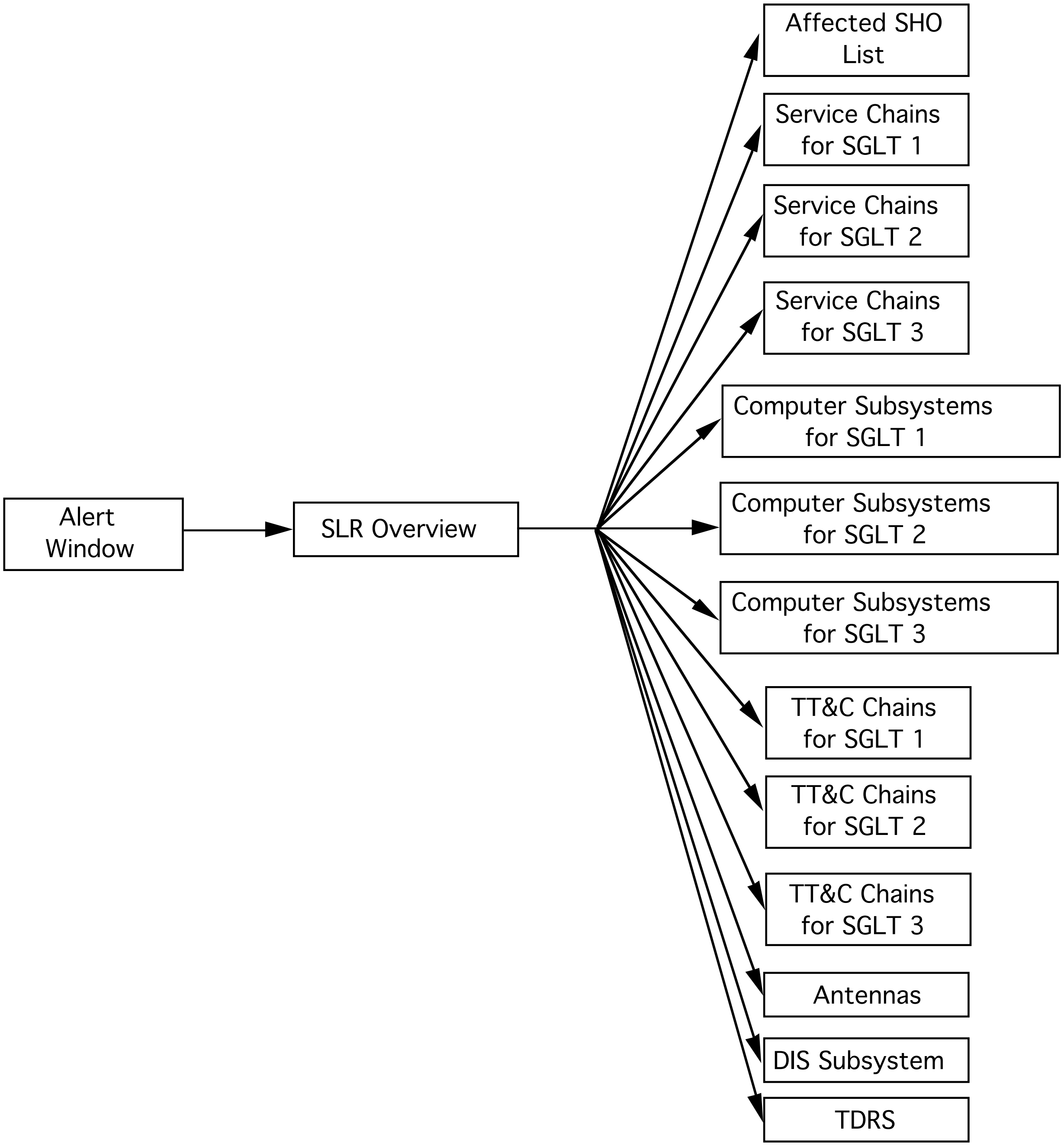

Alert Window
SLR Overview
Affected SHO List
Service Chains for SGLT 1
Service Chains for SGLT 2
Service Chains for SGLT 3
Computer Subsystems for SGLT 1
Computer Subsystems for SGLT 2
Computer Subsystems for SGLT 3
TT&C Chains for SGLT 1
TT&C Chains for SGLT 2
TT&C Chains for SGLT 3
Antennas
DIS Subsystem
TDRS

Exhibit A-2. NCC DS Service Planning Segment Replacement

INDEX